Université de Montréal

Un dictionnaire de régimes verbaux en mandarin

par Linna He

Département de linguistique et de traduction
Faculté des arts et des sciences

Mémoire présenté
en vue de l'obtention du grade de Maître ès arts (M.A.)
en linguistique

Décembre 2020

Université de Montréal
Département de linguistique et de traduction, Faculté des arts et des sciences

---

Ce mémoire intitulé

Un dictionnaire de régimes verbaux en mandarin

Présenté par

Linna He

A été évalué par un jury composé des personnes suivantes

Antoine Venant
Président-rapporteur

François Lareau
Directeur de recherche

Marie-Claude L'Homme
Membre du jury

Résumé

Ce mémoire s'insère dans le projet GenDR, un réalisateur de texte profond multilingue qui modélise l'interface sémantique-syntaxe pour la génération automatique de texte (GAT). Dans le cadre de la GAT, les ressources lexicales sont de première nécessité pour que le système puisse transformer des données nonlinguistiques en langage naturel. Ces ressources lexicales déterminent dans une certaine mesure la précision et la flexibilité des phrases générées. En raison de l'imprévisibilité du régime des verbes et du rôle central que les verbes jouent dans un énoncé, une ressource lexicale qui décrit le régime des verbes revêt une importance particulière pour générer du texte le plus précis et le plus naturel possible.

Nous avons tenté de créer un dictionnaire de régimes verbaux en mandarin. Ce genre de ressource lexicale est toujours une lacune dans le domaine de la GAT en mandarin. En nous basant sur la base de données *Mandarin VerbNet*, nous avons eu recours à Python pour extraire les adpositions régies et créer notre dictionnaire. Il s'agit d'un dictionnaire dynamique, dont le contenu peut être paramétré en fonction des objectifs de l'utilisateur.

Mots-clés : génération automatique de texte; réalisation linguistique; mandarin; verbes; patrons de régime

Abstract

This work fits into the GenDR project, a multilingual deep realizer which models the semantics-syntax interface for natural language generation (NLG). In NLG, lexical resources are essential to transform non-linguistic data into natural language. To a certain extent, the lexical resources used determine the accuracy and flexibility of the sentences generated by a realizer. Due to the unpredictability of verbs' syntactic behaviour and the central role that verbs play in an utterance, a lexical resource which describes the government patterns of verbs is key to generating the most precise and natural text possible.

We aim to create a dictionary of verbs' government patterns in Mandarin. This kind of lexical resource is still missing for NLG in Mandarin. Based on the Mandarin VerbNet database, we used Python to extract information about adpositions and to create our dictionary. This is a dynamic dictionary whose content can be parameterized according to the user's needs.

Keywords : natural language generation; linguistic realization; Mandarin; verbs; government patterns

Table des matières

Liste des tableaux

Liste des figures

Liste des abréviations et des sigles

GAT : Génération automatique de texte

IA : Intelligence artificielle

MV : Mandarin VerbNet

TAL : Traitement automatique des langues

TST : Théorie Sens-Texte

Remerciements

En premier lieu, je voudrais remercier mon directeur de recherche, François Lareau. C'est lui qui m'a soutenue jusqu'au bout de mon mémoire. Il y a une fois où je voulais abandonner ma recherche, c'est son aide qui m'a fait persévérer.

Je voudrais aussi adresser mes remerciements à ma famille. C'est leur soutien qui m'a permis d'étudier à l'étranger.

Je voudrais aussi remercier Xiaoyu Zhao, Shiyu Li, Charlotte Portenseigne et les autres collègues au fil de mes études à l'Université de Montréal.

Je voudrais aussi remercier Jiayu Xie, qui m'a accompagnée pendant ces années solitaires.

Chapitre 1 Introduction

1.1 Problématique

Le traitement automatique des langues (TAL) vise à développer des logiciels capables de traiter de manière automatique des données linguistiques. Le TAL comprend plusieurs sous-domaines, tels que le traitement de la parole, la traduction automatique, la compréhension automatique des textes et la génération automatique de texte (GAT). Notre recherche s'inscrit dans le domaine de la GAT, dont le but est de produire des énoncés en langage naturel à partir de données nonlinguistiques (informatisées).

Un système de GAT est traditionnellement divisé en deux modules majeurs (Danlos, 1983). La première est le *quoi-dire*, *composant stratégique*, *générateur profond*, ou *conceptualiseur*, qui détermine et organiser les informations à transmettre. Le deuxième est appelé *comment-le-dire*, *composant tactique*, *générateur de surface*, ou *formulateur*, qui choisit les unités lexicales avec lesquelles les informations abstraites choisies par le premier module sont formulées en langue naturelle. Gatt et Krahmer (2018) appellent ces deux modules le *early process* et le *late process*.

Notre recherche se situe dans le cadre de la GAT et s'insère dans le projet GenDR (Dubinskaite, 2017; Galarreta-Piquette, 2018; Lambrey, 2017; Lambrey & Lareau, 2016, 2015; Lareau et al., 2018), un réalisateur profond multilingue qui reprend les bases du système MARQUIS (Wanner et al., 2010) et qui opère dans le cadre de la théorie Sens-Texte (TST). Notre objectif initial était de créer un module GenDR spécifique pour le mandarin, qui s'ajouterait aux modules existants (français, anglais, lituanien et persan). Nous travaillons sur la réalisation profonde (interface sémantique-syntaxe) en mandarin, qui correspond à une partie de ce que le deuxième module de GAT fait, la réalisation linguistique. Par rapport aux méthodes statistiques ou neuronales, Lareau et al., (2018) affirment que les méthodes basées sur des règles sont encore meilleures à cet égard, attendu que la précision extrêmement élevée est requise par les systèmes de GAT, pour des applications dans la vie réelle. Par conséquent, nous nous

concentrons sur la réalisation linguistique à base de règles qui consiste à modéliser les connaissances linguistiques d'une langue donnée dans une grammaire et un dictionnaire.

Dans beaucoup de langues, les verbes constituent une partie du discours différente des autres. Comparé à la majorité des parties du discours qui possèdent des régularités à l'égard de leur régime (description syntaxique), il semble qu'il y a plus de variété dans les régimes verbaux. Étant donné l'imprévisibilité du régime des verbes et le rôle central que les verbes jouent dans un énoncé, nous devons admettre que maîtriser les propriétés des verbes est une tâche de la plus haute priorité pour générer du texte le plus précis et le plus naturel possible. C'est toujours un problème non seulement dans le domaine de la GAT, mais aussi dans d'autres domaines du TAL : les grands avantages de la connaissance des régimes verbaux pour le traitement des langues naturelles sont posés par Korhonen et al. (2006); Schuler (2005). Par exemple, PARLER peut s'utiliser comme verbe intransitif (*Linna parle*), transitif indirect (*Linna parle à Charlotte*) ou transitif direct (*Linna parle mandarin*), qui sont trois régimes différents de ce verbe (sans égard au sens). Généralement, ce genre d'information est encodée dans des dictionnaires qui aident les débutants d'une langue à utiliser les verbes et à bien organiser les phrases. En résumé, en ce qui concerne les systèmes de GAT qui doivent réaliser correctement des constructions syntaxiques, il est nécessaire de trouver un dictionnaire qui décrive en détail le régime des verbes d'une langue donnée.

Étant donné cela, afin de créer un module GenDR spécifique pour le mandarin, nous avions besoin d'un dictionnaire de régimes verbaux qui encode de façon exhaustive les constructions verbales du mandarin. Ce genre de dictionnaire de verbes existe pour d'autres langues : VerbNet pour l'anglais (Schuler, 2005), VerbƎnet pour le français (Danlos et al., 2015) et le VerbNet de l'italien (Busso & Lenci, 2016), notamment. Néanmoins, la recherche sur ce domaine pour le mandarin est tellement rare que nous trouvons seulement un dictionnaire, nommé *Mandarin VerbNet*[1] (Liu & Chiang, 2008), qui est une ressource consultable en ligne. Bien que son nom laisse croire qu'il donne les régimes verbaux comme les autres VerbNet, c'est en fait un dictionnaire de verbes basé principalement sur la théorie linguistique *Frame Semantics* (Baker et al., 1998). L'idée centrale de cette théorie est que le sens des lexies doit être décrit en

[1] http://mega.lt.cityu.edu.hk/~yufechen/#/

termes de cadres sémantiques qui décrivent les interactions sémantiques entre la lexie décrite et les participants de la situation dénotée (appelés *frame elements*). Ce dictionnaire contient 16 *Archi Frames*[2] qui se composent de plusieurs *Basic Frames*. Dans ce dictionnaire, les verbes sont classés dans différents cadres en fonction des rôles et des patrons de construction[3]. Nous montrons à la Figure 1 et à la Figure 2 un exemple d'un *Archi Frame* (COMMUNICATION) et un exemple d'une entrée (討論 tǎolùn 'discuter') classée dans cet *Archi Frame*. On voit bien que dans ce dictionnaire chaque entrée (verbe) est décrite avec les *frames* auxquels elle appartient, les *frame elements* en jeu et les patrons de construction (chacun plusieurs phrases illustratives). En bref, nous ne pouvons pas obtenir les propriétés des verbes qui correspondent à notre besoin à partir de ce dictionnaire, puisqu'il ne décrit pas le régime des verbes du mandarin et qu'il ne classe pas les verbes en fonction des comportements syntaxiques comme ce que le VerbNet a fait pour l'anglais.

[2] http://mega.lt.cityu.edu.hk/~yufechen/#/frame-index

[3] http://mega.lt.cityu.edu.hk/~yufechen/#/verb-index

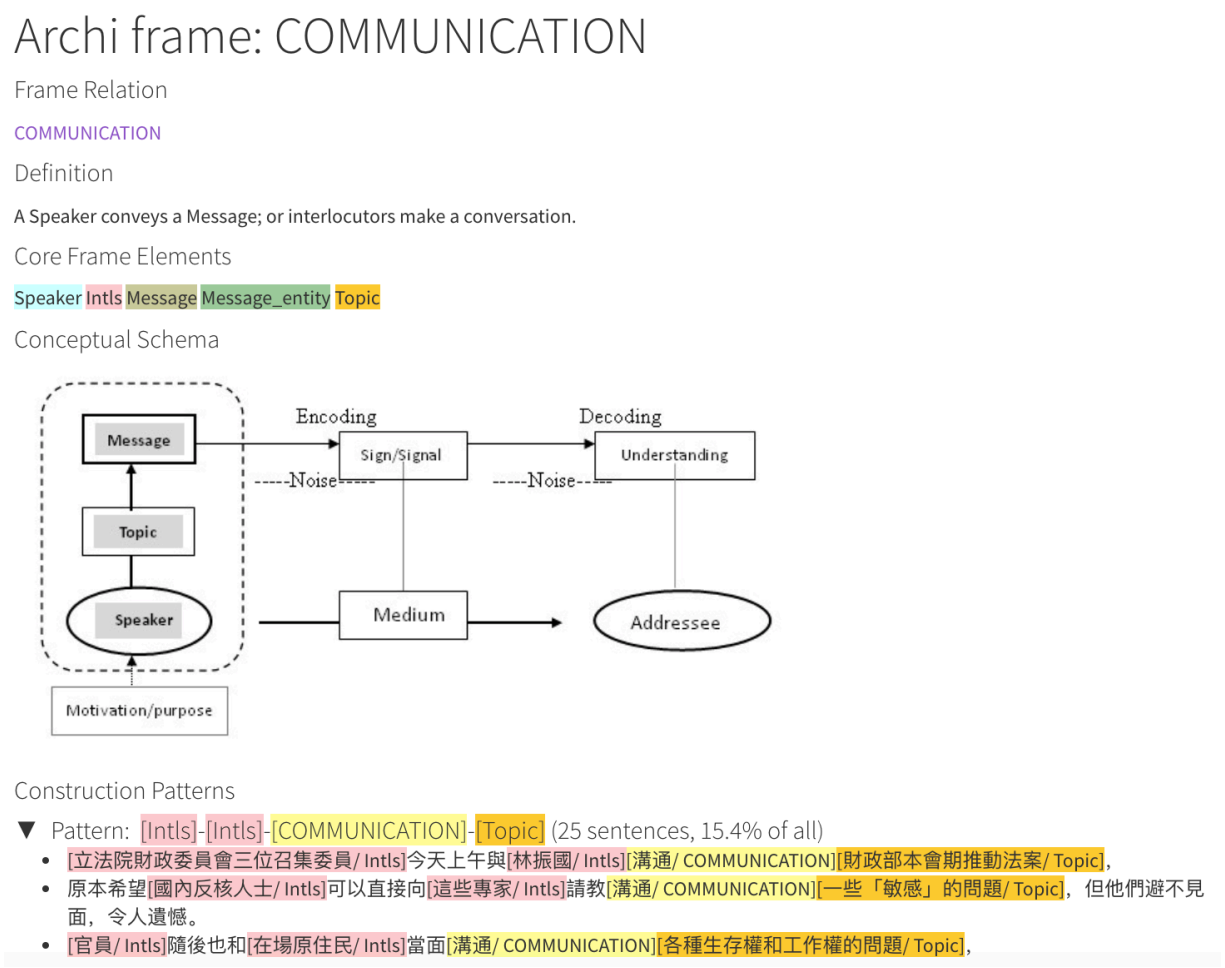


Figure 1. Un exemple d'un *archi frame* dans le dictionnaire *Mandarin VerbNet*

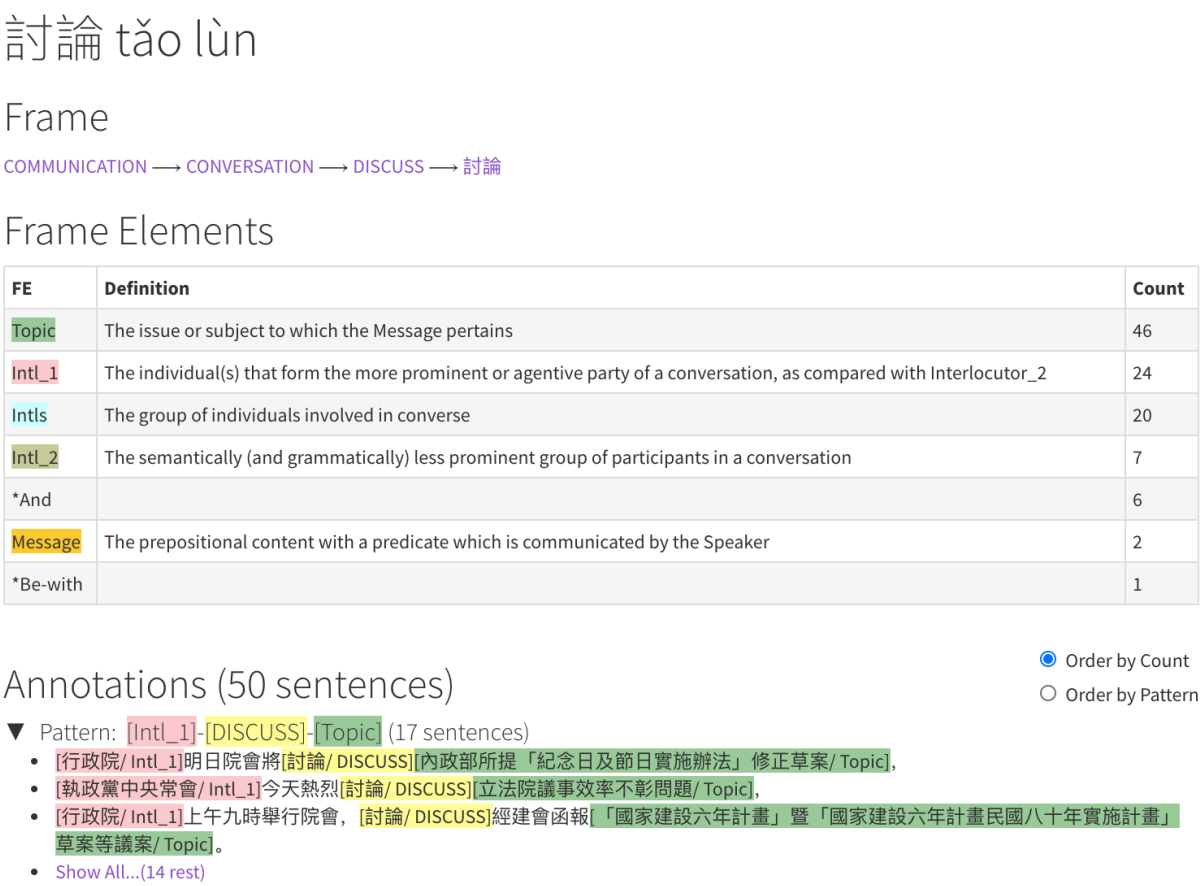


Figure 2. Un exemple d'une entrée (討論 tǎolùn 'discuter') dans le dictionnaire *Mandarin VerbNet*

Ainsi, en vue d'établir une base pour la création d'un module GenDR spécifique pour le mandarin, nous devions créer en premier lieu un dictionnaire de régimes de verbes en mandarin. Ensuite, nous pourrons intégrer ce dictionnaire dans le réalisateur profond GenDR, comme Galarreta-Piquette (2018) l'a fait pour l'anglais, ce qui nous permettra de créer un module GenDR en mandarin. Par ailleurs, ce dictionnaire pourra servir de ressource lexicale pour d'autres études de GAT ou de TAL en mandarin.

1.2 Objectifs du mémoire

Notre recherche vise à créer un dictionnaire de régimes des verbes en mandarin constitué des verbes du dictionnaire *Mandarin VerbNet*. Les buts de cette recherche sont les suivants :

1) Extraire les informations utiles à partir de la base de données de *Mandarin VerbNet*.

2) Analyser les informations extraites puis créer un dictionnaire qui décrit le régime des verbes.

1.3 Organisation du mémoire

Nous avons séparé ce mémoire en six chapitres. Le deuxième chapitre est une mise en contexte. Nous présenterons d'abord la TST, la GAT, le réalisateur profond GenDR, les travaux existants concernant les propriétés des verbes des autres langues et l'intégration de VerbNet dans un réalisateur profond. Ensuite, nous présenterons notre méthodologie de recherche dans le troisième chapitre. Nous essayons deux méthodes : l'une à partir des corpus en mandarin, l'autre à partir d'une ressource existante, *Mandarin VerbNet*. Nous nous servons finalement de la deuxième méthode : nous normalisons le corpus brut de *Mandarin VerbNet* et nous en extrayons l'information des adpositions pour construire un tableau qui décrit les relations entre les verbes et les adpositions régies par ces verbes. Dans le quatrième chapitre, nous présenterons la génération d'un dictionnaire final plus ou moins précis en fonction des besoins pour une application particulière. Au cinquième chapitre, nous ferons une réflexion sur les problèmes rencontrés au cours de la recherche et sur le travail possible à faire dans le futur. Enfin, la conclusion présentera une synthèse de ce mémoire.

Chapitre 2 Mise en contexte

2.1 Théorie Sens-Texte

La théorie Sens-Texte (TST) (Mel'čuk, 1988, 1997; Milićević, 2006; Polguère, 1998) est un cadre théorique linguistique pour la construction de modèles formels des langues naturelles. Elle propose une partition de la modélisation d'un énoncé en quatre niveaux de représentation majeurs : sémantique, syntaxique, morphologique et phonologique. La TST est basée sur les trois postulats suivants (Mel'čuk, 1997).

Postulat 1

> Une langue est un système fini de règles qui décrit la correspondance entre un ensemble infini de sens et un ensemble infini de textes.

Les sens et les textes sont tous représentés par des objets formels appelés, respectivement, les représentations sémantiques (RSém) et les représentations phonétiques (RPhon).

Postulat 2

> La correspondance Sens-Texte doit être décrite par un dispositif logique, ou un système de règles, qui constitue un modèle linguistique fonctionnel : un modèle Sens-Texte.

Un modèle Sens-Texte produit en sortie un ensemble de textes (RPhon) à partir des sens (RSém), ce qui ressemble à l'activité linguistique du locuteur natif. Les textes produits contiennent toutes les paraphrases pouvant exprimer le sens donné. Polguère (1998) voit ces modèles comme des machines logiques virtuelles, comme illustré à la Figure 3.

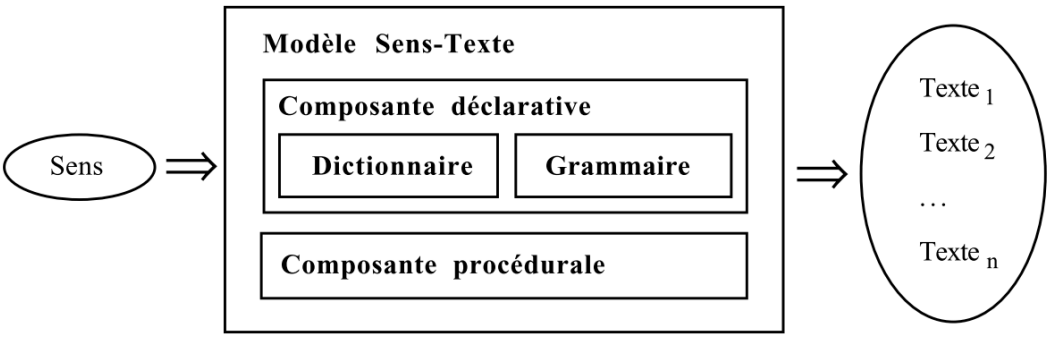


Figure 3. La structure fonctionnelle d'un modèle Sens-Texte, tirée de (Polguère, 1998)

Postulat 3

> Pour décrire la correspondance Sens-Texte, deux niveaux intermédiaires de représentation des énoncés sont nécessaires pour mettre en lumière les faits linguistiques pertinents : la représentation syntaxique (RSynt), qui correspond aux régularités spécifiques à la phrase, et la représentation morphologique (RMorph), qui correspond aux régularités spécifiques au mot.

Parmi les quatre niveaux de représentation majeurs, les trois derniers sont subdivisés en deux sous-niveaux : le sous-niveau profond, qui est orienté vers le sens et le sous-niveau de surface, qui est orienté vers le texte. La Figure 4 illustre la structure du modèle Sens-Texte qui se compose de six composantes successives contenant des règles pour faire le passage d'une représentation à la représentation prochaine.

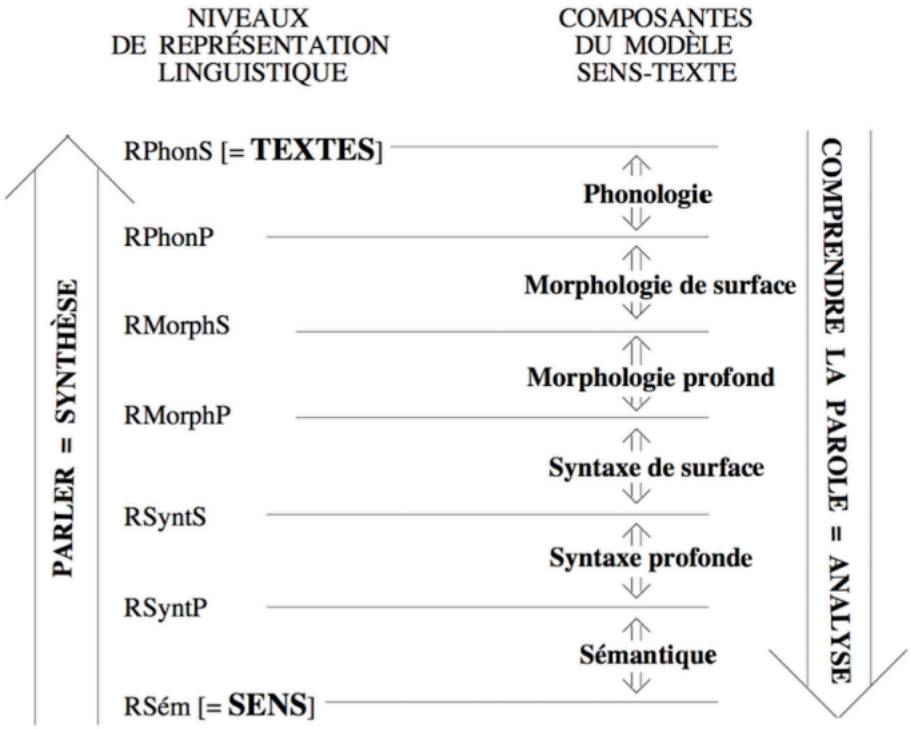


Figure 4. La structure du modèle Sens-Texte, tirée de (Mel'čuk, 1997)

Pour rendre plus clair ce modèle, nous présentons ci-dessous, du sens vers le texte, la structure de base de chaque représentation en donnant des exemples. Avant tout, il y a quelques définitions à présenter pour une meilleure compréhension de la présentation des représentations.

Définition : Sémantème

> L'unité sémantique de base, appelée sémantème, est un sens lexical, c'est-à-dire le signifié d'une unité lexicale (= lexie) de la langue. (Mel'čuk & Milićević, 2014, p. 109)

Définition : Lexème

> Un lexème est un ensemble de mots-formes et de syntagmes de type particulier (= formes analytiques) dont les signifiés ne se distinguent que par des significations flexionnelles et dont les signifiants incluent le signifiant du même radical, qui exprime leur signification partagée. (Mel'čuk & Milićević, 2014, p. 250)

Définition : Lexie

> Une lexie est soit un lexème, soit une locution. (Mel'čuk & Milićević, 2014, p. 251)

Définition : Lexie profonde

> Une lexie profonde est soit une lexie véritable (sémantiquement) pleine, soit le nom d'une fonction lexicale, soit encore une lexie fictive. (Mel'čuk & Milićević, 2014, p. 181)

Définition : Lexie (sémantiquement) pleine

> Une lexie pleine a un correspondant direct dans la structure sémantique : un sémantème ou une configuration de sémantème. Une telle lexie est soit un lexème soit une locution. (Mel'čuk & Milićević, 2014, p. 181)

La représentation sémantique (RSém), dont la structure de base est un réseau sémantique, dans lequel les sens sont représentés par des nœuds étiquetés de sémantèmes et les relations prédicat-argument par des arcs numérotés pour distinguer les arguments d'un même prédicat. La Figure 5 décrit un réseau sémantique un peu complexe qui correspond aux phrases ci-dessous; la figure et les phrases proviennent de (Mel'čuk, 2015).

1) *Damas a permis à Abou-Kalaf d'augmenter l'afflux de terroristes en Irak à trente par mois.*
2) *Le gouvernement Syrien a laissé Abou-Kalaf accroître la quantité de terroristes pénétrant en Irak jusqu'à trente chaque mois.*
3) *Abou-Kalaf a eu le feu vert des dirigeants syriens pour faire monter le nombre des militants d'Al-Qaeda qui affluent en Irak à trente par mois*.

Par ailleurs, il y a trois autres composantes d'une RSém que nous ne discutons pas ici : la structure sémantico-communicative, la structure rhétorique et la structure référentielle. Pour plus d'informations, voir (Mel'čuk, 2012).

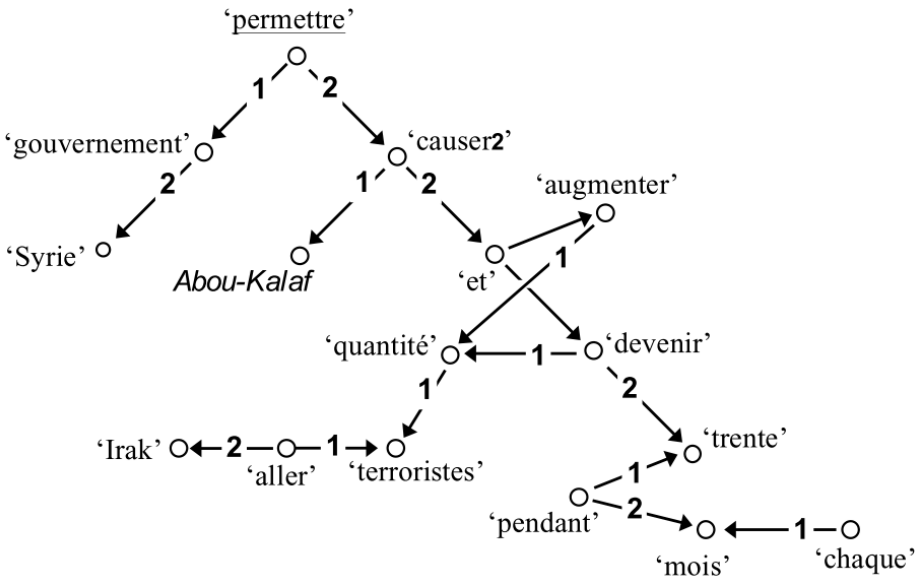


Figure 5. Une structure sémantique, tirée de (Mel’čuk, 2015)

La représentation syntaxique profonde (RSyntP) est un arbre de dépendances non linéairement ordonné dans lequel il n’existe que des lexies pleines (représentées par les nœuds) liées par deux types principaux de relations : complémentation (étiquetée en chiffres romains) et modification (étiquetée ATTR). L’exemple à la Figure 6 est seulement une de plusieurs RSyntP qu’on peut produire à partir de la RSém ci-dessus.

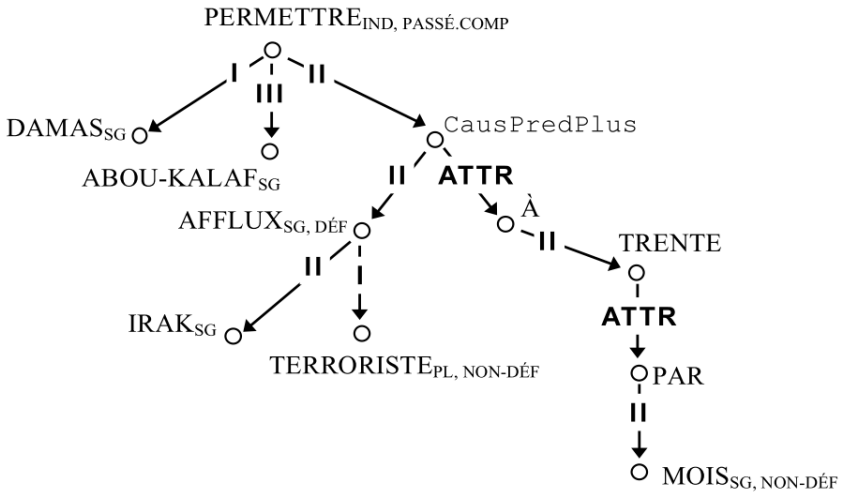


Figure 6. Une structure syntaxique profonde, tirée de (Mel’čuk, 2015)

La représentation syntaxique de surface (RSyntS) est aussi un arbre de dépendance non linéairement ordonné, dans lequel les nœuds représentent des lexies pleines ou vides et les arcs des relations syntaxiques de surface (spécifiques à chaque langue). La Figure 7 illustre la structure syntaxique de surface qui correspond à la RSyntP ci-dessus.

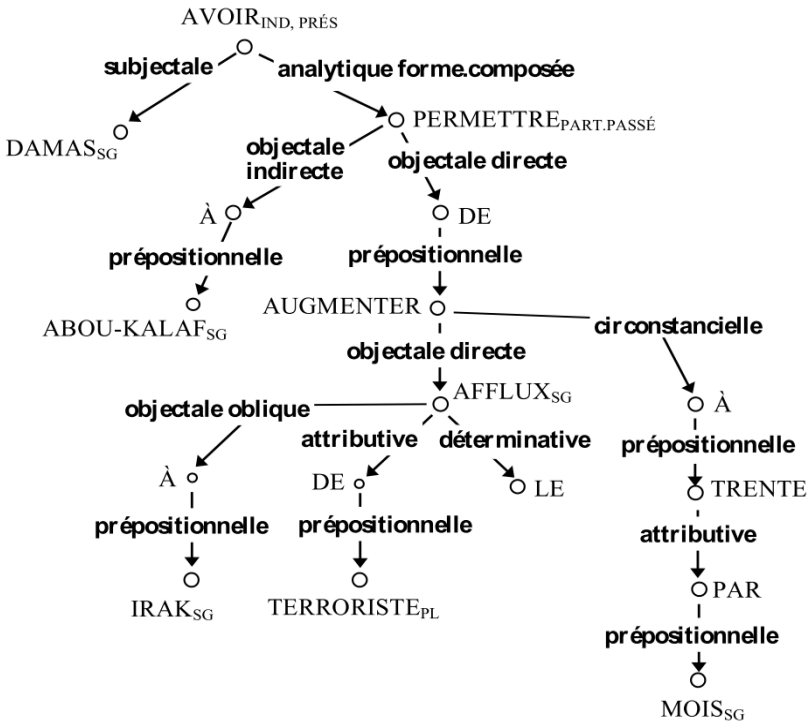


Figure 7. Une structure syntaxique de surface, tirée de (Mel'čuk, 2015)

La représentation morphologique profonde (RMorphP) est une chaîne de lexèmes munis de toutes les valeurs de leurs catégories flexionnelles (grammèmes).

DAMAS$_{SG}$ AVOIR$_{IND, PRÉS, 3, SG}$ PERMETTRE$_{PART_PASSÉ}$ À ABOU-KALAF$_{SG}$ ||
DE AUGMENTER$_{INF}$ |
LE$_{MASC, SG}$ AFFUX$_{SG}$ DE TERRORISTE$_{PL}$ EN IRAK$_{SG}$ |
À TRENTE PAR MOIS$_{SG}$ |||

Figure 8. Une structure morphologique profonde, tirée de (Mel'čuk, 2015)

La représentation morphologique de surface (RMorphS) est une chaîne de groupes de morphèmes, qui correspondent chacun à un mot-forme.

{DAMAS}⊕{SG} {AVOIR}⊕{IND.PRÉS}⊕{3.SG} {PERMETTRE}⊕{PART_PASSÉ} {À} {ABOU-KALAF}⊕{SG} ||
{DE} {AUGMENTER}⊕{INF} |
{LE}⊕{MASC}⊕{SG} {AFFLUX}⊕{SG} {DE} {TERRORISTE}⊕{PL} {EN} {IRAK}⊕{SG} |
{À} {TRENTE} {PAR} {MOIS}⊕{SG} |||

Figure 9. Une structure morphologique de surface, tirée de (Mel'čuk, 2015)

La représentation phonologique profonde (RPhonP) contient deux structures : la structure phonologique profonde et la structure phonique-prosodique profonde. Nous ne présentons ici que la première structure, la structure de base. Pour plus d'informations sur la structure phonique-prosodique profonde, voir (Mel'čuk & Milićević, 2014). Écrite en transcription phonologique, la structure phonologique profonde est une chaîne de phonèmes constituant la phrase, qui ne contient pas les sandhis (modifications phonétiques qui ont lieu à la frontière entre deux morphèmes à l'intérieur d'un mot-forme ou à la frontière entre deux mots-formes dans un énoncé, par exemple, la liaison, l'élision, etc.). La Figure 10 illustre la structure phonologique profonde qui suit la RMorphS présentée ci-dessus.

/damas#a#pɛʁmi#a#abukalaf#də#ɔgmɑ̃te#lə#afly#də#tɛʁɔʁist#ɑ̃#iʁak#a#tʁɑ̃t#paʁ#mwa/

Figure 10. Une structure phonologique profonde, tirée de (Mel'čuk, 2015)

La représentation phonologique de surface (RPhonS) comporte aussi une paire de structures : la structure phonologique de surface et la structure phonique-prosodique de surface. En tant que structure de base, la structure phonologique de surface est une chaîne de phones; elle est écrite en transcription phonétique, comme illustré à la Figure 11. Pour plus d'informations sur la structure phonique-prosodique de surface, voir (Mel'čuk & Milićević, 2014).

[damasapɛʁmiaabukalafdɔgmɑ̃telaflydətɛʁɔʁistɑ̃niʁakatʁɑ̃tpaʁmwa]

Figure 11. Une structure phonologique de surface

Après les passages successifs, la phrase correspondante est synthétisée (*Damas a permis à Abou-Kalaf d'augmenter l'afflux de terroristes en Irak à trente par mois.*). Ce sont les règles constituant chaque module correspondant qui permettent la synthèse. Ainsi, on retrouve trois

classes majeures de règles linguistiques dans un modèle Sens-Texte : règles sémantiques, syntaxiques et morphologiques.

Les règles sémantiques accomplissent la tâche du module sémantique de produire toutes les RSyntP synonymes qui correspondent à une RSém donnée. À l'intérieur du module sémantique, les règles sont divisées en deux types : règles de transition et règles de paraphrasage. Nous ne parlons ici que des règles de transition; pour plus d'informations sur les règles de paraphrasage, voir (Mel'čuk & Milićević, 2014; Milićević, 2007). À la Figure 6, nous avons vu que la structure syntaxique profonde nous montre trois types d'unités linguistiques : lexies profondes, relations syntaxiques profondes et grammèmes profonds. En fonction des trois, les règles sémantiques de transition se divisent en deux sous-types de règles (Mel′čuk, 2007) :

1) Règles de lexicalisation se subdivisent en deux types de règle :

   - Règles lexicales, qui traitent les configurations de sémantèmes en leur associant des lexies profondes. Par exemple, les règles lexémiques établissent la correspondance entre les configurations sémantiques et les lexèmes.

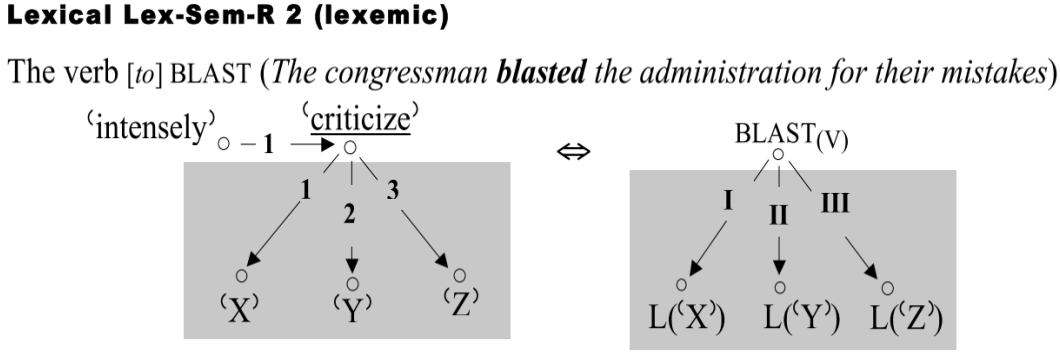


Figure 12. Un exmple d'une règle lexémique, tiré de (Mel′čuk, 2007)

   - Règles flexionnelles, qui traitent les configurations de sémantèmes en leur associant des grammèmes profonds et des dérivatèmes, ou significations dérivationnelles. Par exemple, certaines règles flexionnelles associent les grammèmes pleins aux lexies profondes.

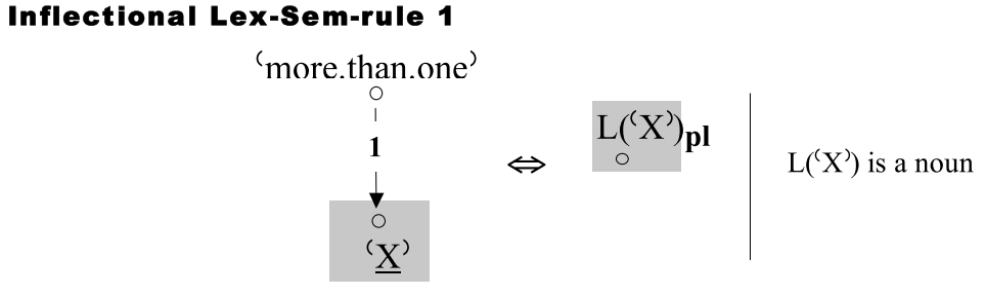


Figure 13. Un exemple d'une règle flexionelle, tiré de (Mel′čuk, 2007)

2) Règles d'arborisation, qui traitent les relations sémantiques en leur associant des relations syntaxiques profondes. Un type de règles d'arborisation établissent le sommet de l'arbre syntaxique profond et d'autres consistent à construire des branches et des sous-arbres de l'arbre syntaxique profond.

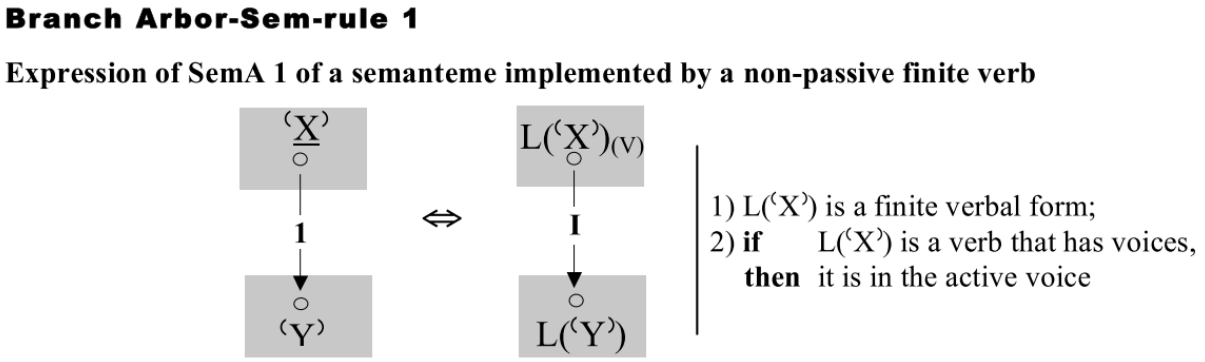


Figure 14. Un exemple d'une règle de branche, tiré de (Mel′čuk, 2007)

Les règles syntaxiques sont divisées en deux groupes correspondant à deux sous-modules syntaxiques : règles syntaxiques profondes, qui établissent la correspondance entre la RSyntP et la RSyntS, et règles syntaxiques de surface, qui établissent la correspondance entre la RSyntS et la représentation morphologique profonde de la phrase (RMorphP). Les règles syntaxiques profondes de transition accomplissent les trois tâches suivantes (Mel'čuk & Milićević, 2014) :

1) Faire correspondre les éléments du lexique profond et les relations syntaxiques profondes aux éléments du lexique de surface et aux relations syntaxiques de surface.

2) Faire le passage des grammèmes du niveau syntaxique profond au niveau syntaxique de surface.

3) Exécuter la pronominalisation et l'effacement de certains éléments lexicaux de la structure syntaxique profonde.

Les règles syntaxiques de surface contiennent trois classes majeures de règles : les règles traitant la linéarisation (qui déterminent l'ordre linéaire des lexèmes), les règles traitant la prosodisation (qui associent les prosodies à des lexèmes), et les règles traitant la morphologisation (qui calculent et ajoutent les grammèmes syntaxiques d'un lexème).

Les règles morphologiques, au sein du module morphologique qui se subdivise en deux sous-modules, sont aussi regroupées en deux types de règles : règles morphologiques profondes, qui assurent le passage de la RMorphP à la RMorphS, et règles morphologiques de surface, qui permettent le transition de la RMorphS à la représentation morphique (RMorphique). Il faut mentionner que le sous-module morphologique de surface comprend également les règles morphonologiques pour effectuer le transition de la RMorphique à la RPhonP, voir (Mel'čuk & Milićević, 2014).

Jusqu'ici, nous avons fait une introduction générale de la TST. Comme elle est le cadre théorique du projet GenDR, plus de détails vont être présentés à la section sur GenDR (cf. §2.3).

2.2 Génération automatique de texte

La génération automatique de texte (GAT) est une branche du traitement automatique des langues (TAL) qui vise à produire du texte compréhensible en langue naturelle à partir de données nonlinguistiques (informatisées). Considérée comme le contraire de la compréhension du langage naturel, qui doit désambiguïser la phrase d'entrée pour produire une représentation abstraite, la GAT doit décider comment mettre un concept en mots. Les techniques de GAT vont des systèmes simples basés sur des patrons, comme le publipostage qui génère des lettres à partir d'un modèle rigide, aux systèmes basés sur une modélisation complexe de la langue. La GAT peut également être réalisée par un modèle utilisant l'apprentissage automatique (un champ d'étude de l'intelligence artificielle), généralement entraîné sur un grand corpus.

Les systèmes de GAT sont le plus souvent utilisés pour alléger la charge de travail des humains : des documents courants, comme des bulletins météorologiques et des lettres commerciales, sont produits automatiquement. Les premiers systèmes commerciaux de conversion de données en texte produisaient des prévisions météorologiques à partir de données météorologiques. Le premier système de ce type à être mis en valeur était FoG (*Forecast Generator*), utilisé par Environnement Canada pour produire des prévisions météorologiques en français et en anglais (Goldberg et al., 1994). Les systèmes de GAT sont également employés comme outils d'explication interactifs pour communiquer des informations de manière compréhensible à des utilisateurs non experts, en particulier dans les domaines de l'ingénierie logicielle (Rambow et Korelsky, 1992) et de la médecine (Carenini, Mittal et Moore, 1994). Récemment, la GAT est de plus en plus appliquée dans les technologies commerciales, notamment pour les agents virtuels comme Siri (développé par *Apple*) et Alexa (développé par *Amazon*). Ce genre d'application révolutionne la manière dont les professionnels et les consommateurs interagissent avec les ordinateurs. Par ailleurs, d'un point de vue linguistique, la GAT offre aux linguistes une méthode qui leur permet de tester leurs théories linguistiques et de vérifier si leur modélisation de la langue contient des erreurs (Danlos, 1983; Lareau, 2002).

2.2.1 Architecture d'un système de GAT

La production d'un texte consiste essentiellement en deux types de décisions : décisions conceptuelles et décisions linguistiques. Le premier type de décisions concerne des problèmes comme : quelles sont les informations à exprimer? Comment organiser les informations à transmettre? Le deuxième type de décisions portent sur des problèmes comme : quelles unités lexicales et quelles constructions syntaxiques choisir? Comment diviser le texte en phrases et en paragraphes (Danlos, 1983)? Par conséquent, Danlos (1983) divise de manière grossière un système de GAT en deux modules majeurs : le *quoi-dire* et le *comment-le-dire*, qui correspondent aux *early process* et *late process* de Gatt et Krahmer (2018). Thompson (1977) a fait aussi une distinction entre les décisions de *quoi-dire* et *comment-le-dire* en les appelant respectivement *strategic component* pour le contenu et la structure du texte, et *tactical component* pour la morphologie et la grammaire. Bateman et Zock (2003) indiquent que le problème majeur de GAT est le choix qui se pose à plusieurs niveaux. Les choix de contenu

correspondent au module *quoi-dire*, et les choix lexicaux et syntaxiques au module *comment-le-dire*.

Reiter et Dale (2000) ont cherché à dégager une architecture consensuelle des différents systèmes de GAT. Ils observent que les systèmes de GAT peuvent se découper en six modules :

1) Sélection du contenu : décider quelle information à communiquer dans le texte.
2) Structuration du document : organiser l'information dans une structure rhétoriquement cohérente.
3) Agrégation : décider comment diviser les phrases en ajoutant des dispositifs de cohésion.
4) Lexicalisation : décider quelles unités lexicales utiliser pour exprimer le contenu sélectionné.
5) Génération d'expressions référentielles : décider quelles expressions utiliser pour faire référence à des entités.
6) Réalisation linguistique : produire les phrases individuelles grammaticalement correctes.

L'entrée est une représentation logique et la sortie est son expression en langue naturelle. Les trois premiers modules font partie du *quoi-dire*, et les trois derniers forment le *comment-le-dire*. Nous allons expliquer de manière détaillée chacune des étapes ci-dessous pour mieux comprendre le processus séquentiel qu'un système de GAT effectue.

Ce que nous faisons dans la première étape, sélection du contenu, c'est de déterminer l'information à communiquer dans le texte. Il est important pour un système de GAT de distinguer les informations qui sont pertinentes à être transmises dans le texte de celles qui ne le sont pas. Le choix du contenu à exprimer dans un texte dépend de nombreux facteurs : objet de communication, objectif de communication, etc. Par exemple, on ne peut pas présenter la même information d'un domaine donné à un expert qu'à un profane, puisque ce dernier a besoin de beaucoup plus d'informations explicatives que le premier. Des objectifs de communication différents exigent également des informations différentes à transmettre. Par exemple, si un système de GAT est capable à la fois de produire le compte rendu d'un match de basket-ball et

de fournir des définitions et des explications des règles et des termes du domaine, un contenu tout à fait différent sera requis dans les deux cas.

Ensuite, la structuration du document est l'étape où nous décidons dans quel ordre les informations sélectionnées dans la première étape seront présentées. Cette étape consiste à transformer le contenu sélectionné en représentation ordonnée et structurée. Par exemple, si nous nous servons du système de GAT mentionné dans le paragraphe précédent pour générer un compte rendu d'un match de basket-ball, le texte devra commencer par les informations générales à propos du match (p. ex., lieu et date du match), suivi des informations sur les équipes qui participent au match (p. ex., noms et joueurs principaux des équipes), puis des informations sur le déroulement du match (faits saillants du match), et finalement des informations liées au résultat du match (score final, gagnant du match). Cet exemple est inspiré de (Galarreta-Piquette, 2018).

La troisième étape, l'agrégation, sert à combiner les messages sélectionnés et structurés à la deuxième étape, puisque nous ne pouvons pas les exprimer dans des phrases individuelles. Il s'agit de décider comment séparer les informations en phrases et en paragraphes individuels et d'ajouter des dispositifs de cohésion (pronoms, marqueurs de discours), le cas échéant. En un mot, cette étape a pour but de réduire la redondance et de rendre le texte plus fluide et lisible. L'exemple suivant illustre le fonctionnement de cette étape : les messages sélectionnés et structurés par les deux premières étapes (*Paul est entré dans la cuisine. Paul a allumé le four. Paul a mis le gâteau dans le four.*) s'agrègent à la troisième étape en un texte cohérent (*Paul est entrée dans la cuisine, a allumé le four, et y a mis le gâteau.*).

La lexicalisation est l'étape où nous commençons à transformer les données non linguistiques en langue naturelle. Il s'agit de choisir les unités lexicales pour exprimer le contenu sélectionné. Les langues permettent d'exprimer un même sens de plusieurs manières, notamment en utilisant diverses combinaisons d'unités lexicales (c'est le pouvoir de paraphrasage de la langue). Pour cette raison, le choix des lexies est une étape assez complexe. Reiter et Dale (2000) incluent également le choix d'autres moyens d'expression linguistiques, tel que des constructions syntaxiques. Par exemple, en français, la possession peut être exprimée soit lexicalement (*le livre appartient à Linna*), soit par la construction possessive (*le livre de Linna*). Dans leur architecture consensuelle, la lexicalisation se fait avant la réalisation. Il faut

noter que dans la réalisation profonde (interface sémantique-syntaxe) sur laquelle nous travaillons, la lexicalisation et le choix de la structure syntaxique se font en même temps, c'est-à-dire que la réalisation linguistique pour Reiter et Dale (2000) n'est pas la même chose que dans le réalisateur profond GenDR. Nous allons l'expliquer plus tard (cf. §2.2.2).

La génération d'expressions référentielles est l'étape où nous choisissons la façon de faire référence à une entité qui permette au lecteur d'identifier correctement cette entité dans un contexte donné. Comme la lexicalisation, la génération d'expressions référentielles est également un problème pour la GAT, car une même entité peut être identifiée de différentes manières. Par exemple, toujours dans le domaine du basket-ball, on peut faire référence au joueur Ming Yao de diverses manières dans le texte : *Ming Yao*, *Big Yao*, *Ming Dynasty*, *Chairman Yao*, *Shaquie Chan*, *The Great Wall of China*.

La dernière étape est la réalisation linguistique, qui consiste à appliquer un ensemble de règles qui définissent une langue naturelle donnée à une représentation plus abstraite afin de produire un texte morphologiquement, syntaxiquement et typographiquement correct (p. ex., mise en majuscule du premier mot d'une phrase et ponctuation). Comme nous l'avons mentionné plus haut, la réalisation de Reiter et Dale (2000) et celle qui s'effectue dans GenDR ne sont pas la même chose. Nous allons donc expliquer cette différence et présenter de manière détaillée la réalisation linguistique à laquelle s'attache notre travail dans la partie suivante.

2.2.2 Réalisation linguistique

Dans cette partie, nous présenterons la dernière étape dans un système de GAT, la réalisation linguistique, à laquelle a rapport notre recherche. Nous allons distinguer deux types de réalisation, expliquer trois méthodes et présenter brièvement quelques réalisateurs existants.

Avant tout, nous distinguons deux types de réalisation : la réalisation de surface et la réalisation profonde. La réalisation de surface se fait à partir de structures syntaxiques lexicalisées (entrée plus proche du texte). Ce type de réalisation correspond exactement à ce que nous avons décrit dans l'architecture consensuelle d'un système de GAT décrite par Reiter et Dale (2000). Comme ce que nous avons présenté, la lexicalisation est suivie de la réalisation dans la réalisation de surface. La réalisation profonde se fait à partir de structures sémantiques, plus abstraites que des structures syntaxiques. C'est ce type de réalisation qui s'exécute dans le

réalisateur profond GenDR. Cette fois-ci, la lexicalisation et la réalisation sont intimement liées et s'effectuent ensemble à l'intérieur de la réalisation profonde.

En général, il y a trois types de méthodes pour la réalisation. Premièrement, les **méthodes à base de patrons**, où le texte se réalise de deux manières principales : à partir de phrases à trous et à partir de modèles par concaténation de bouts de phrases (des syntagmes ou des propositions complètes). La première fait référence à des textes presque complets dans lesquels il y a des trous à remplir avec des variables. Dans la deuxième méthode, les phrases sont formées en mettant plusieurs bouts de phrases (éléments que l'on veut dire) ensemble. C'est par exemple la technique utilisée par le site de réservations Booking.com. Les textes sont plus variés que ceux du premier type mais il y a encore des structures préétablies, puisqu'on a des structures plus petites que la phrase que l'on recombine. Ainsi, les textes générés par cette méthode font preuve d'une variation linguistique réduite. Mais, en même temps, cette méthode diminue les erreurs puisqu'on a plus de contrôle sur le texte final. Étant donné ses limites, cette méthode est généralement utilisée dans des domaines très spécifiques pour produire des textes récurrents.

Deuxièmement, les **méthodes par apprentissage automatique**. Certaines méthodes de ce type sont basées sur des calculs statistiques. Un système de GAT qui se sert de ce type de méthode énumère tous les outputs possibles pouvant être générés à partir de la représentation linguistique abstraite (sémantique). Les outputs alternatifs seront évalués à l'aide de techniques statistiques qui préfèrent les outputs consistant en combinaisons de mots qui correspondent le mieux à celles observées dans un texte réel. Les grands corpus textuels offrent la possibilité d'appliquer les méthodes statistiques à la GAT, puisque cette méthode repose sur des exemples observables et récurrents de phénomènes linguistiques. Ce que Langkilde (2000) a fait dans son travail pour la génération de phrases est un exemple de système de ce genre. Il y a aussi des méthodes neuronales où on utilise des réseaux de neurones artificiels, qui sont de puissants modèles d'apprentissage. L'apprentissage profond, faisant partie d'une famille plus large de méthodes d'apprentissage automatique, permet à l'ordinateur de s'entraîner lui-même à traiter des données. Les méthodes d'apprentissage profond ont récemment obtenu un grand succès empirique en matière de traduction automatique, de génération de réponses en dialogue, et autres tâches de génération de texte. Certains réseaux de neurones fonctionnent comme un encodeur

qui prend en input des représentations de notre monde (p. ex., une image ou une phrase écrite) et les encode dans un espace vectoriel. D'autres réseaux de neurones fonctionnent comme un décodeur qui transforme les vecteurs en langage naturel. Un réseau de neurones est organisé en trois types de couches : couche d'input, couche(s) cachée(s) et couche d'output. Dans le domaine de la GAT, les réseaux de neurones récurrents (Elman, 1990) sont couramment utilisés. Ces derniers sont une variante d'un réseau de neurones récursif dans lequel les connexions entre neurones forment un cycle dirigé. Cela signifie que l'output dépend non seulement des inputs actuels, mais aussi de l'état des neurones de l'étape précédente, ce qui simule une mémoire. Cette mémoire permet de résoudre des problèmes de TAL et de GAT. Tarasov (2015) présente un modèle de réseau de neurones récurrent qui est capable de générer des paraphrases d'une phrase en input dans la même langue, et des résumés de documents.

Troisièmement, les **méthodes basées sur des modèles linguistiques symboliques** consistent à représenter le langage à travers des règles formelles et des dictionnaires. Le développement de ces modèles exige souvent beaucoup de temps et de ressources, car cela se fait manuellement. En comparant les méthodes statistiques et celles basées sur des règles, nous nous demandons si les méthodes statistiques prennent le dessus, puisque celles-ci économisent à la fois du temps et les ressources humaines. Pour mieux comparer les deux types de méthode, nous présentons le problème d'évaluation dans la GAT. Deux approches d'évaluation sont utilisées dans ce domaine : évaluation humaine (Mellish et Dale, 1998) et évaluation automatique basée sur des corpus (comparer les textes générés par le système de GAT à un corpus de textes humains). Belz et Kow (2009) réfléchissent sur la question : est-ce que le coût réduit de construction du système et l'adaptabilité accrue sont obtenus au prix d'une réduction de la qualité de l'output, et si oui, à quel point? Ils utilisent les deux approches d'évaluation mentionnées ci-dessus pour évaluer de manière comparative la qualité des outputs de dix systèmes, parmi lesquels quatre créés principalement par des méthodes statistiques et six du même domaine créés principalement par des méthodes basées sur des règles. Ils constatent que les évaluations humaines penchent pour les systèmes à base de règles et que les métriques d'évaluation automatique, comme BLEU (Papineni et al., 2002), sous-estiment la qualité des systèmes à base de règles et surestiment la qualité des systèmes créés automatiquement. À ce sujet, Lareau et al. (2018) soutiennent que les méthodes basées sur des règles sont toujours utiles,

en raison de la précision extrêmement élevée requise pour les systèmes de GAT dans plusieurs applications concrètes. Dans notre travail, nous nous concentrons sur la réalisation linguistique à base de règles.

Finalement, nous présenterons en abrégé trois réalisateurs de surface, puis deux réalisateurs profonds.

RealPro (Lavoie et Rainbow, 1997) est un réalisateur de surface basé sur la TST (cf. §2.1). RealPro est implémenté en C++ et il prend en input une structure syntaxique profonde (RSyntP dans la TST). Cette représentation est un arbre non ordonné avec des nœuds étiquetés avec des lexèmes pleins de la langue cible et des arcs étiquetés avec des relations syntaxiques universelles. À l'intérieur de RealPro, chaque transformation est traitée par un module séparé. La Figure 15 illustre l'architecture du système. On peut voir qu'il y a deux grammaires qui servent, respectivement, à transformer une représentation profonde en représentation de surface (*DSynt Grammar*) et à transformer celle-ci en structure morphologique profonde (*SSynt Grammar*). Ce réalisateur a besoin aussi d'un dictionnaire (*Lexicon*) qui encode les propriétés lexicales et morphosyntaxiques des unités pour assurer le fonctionnement de tous les modules. La Figure 16 illustre l'input (RSyntP) pour la phrase *The month was cool and dry with the average number of rain days*.

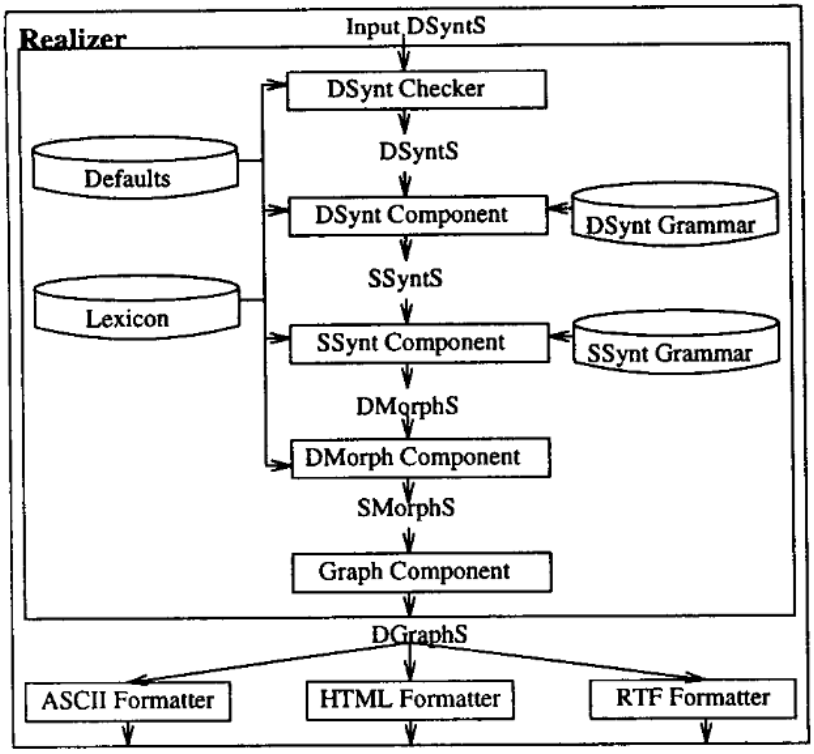


Figure 15. Architecture du système RealPro, tiré de (Lavoie et Rainbow, 1997)

```
BE1 [tense:past]
    (I month [class:common-noun article:def]
     II cool [class:adjective]
          (COORD AND2 []
             (II dry [class:adjective]))
     ATTR WITH1
        (II number [class:common-noun article:def]
             (ATTR average [class:adjective]
              ATTR OF1
                 (II day [class:common-noun number:pl article:no-art]
                      (ATTR rain [class:common-noun])))))
```

Figure 16. Un input RSyntP dans RealPro (*The month was cool and dry with the average number of rain days*), tiré de (Reiter et Dale, 2000)

SimpleNLG est un réalisateur de surface très utilisé. Écrit en Java, sa première version était pour l'anglais, mais il a ensuite été adapté pour d'autres langues, notamment l'allemand (Bollmann, 2011), le français (Vaudry & Lapalme, 2013), le portugais (de Oliveira & Sripada, 2014), l'italien (Mazzei et al., 2016) et l'espagnol (Ramos-Soto et al., 2017). SimpleNLG prend

en input une structure syntaxique déjà lexicalisée encodée en XML. Il effectue en quatre étapes pour la réalisation de textes :

1) Initialiser la structure d'input en mettant les lexèmes dedans en correspondance avec leurs entrées de dictionnaire.
2) Déterminer les traits spécifiques des lexèmes.
3) Combiner les lexèmes en structures plus grandes, jusqu'à un syntagme phrastique.
4) Appliquer les règles morphologiques et accorder les lexèmes pour obtenir les formes fléchies.

Comme SimpleNLG est un réalisateur basé sur des règles, il fait appel à une grammaire, qui contient des règles qui modélisent la syntaxe et la morphologie, et un dictionnaire qui encode les propriétés syntaxiques et morphologiques des unités lexicales. La Figure 17 illustre l'input encodé en XML pour réaliser la phrase *Refactoring is needed*.

```
<Document>
  <child xsi:type="SphraseSpec">
    <subj xsi:type="VPPhraseSpec" FORM="PRESENT_PARTICIPLE">
      <head cat="VERB">
        <base>refactor</base>
      </head>
    </subj>
    <vp xsi:type="VPPhraseSpec" TENSE="PRESENT">
      <head cat="VERB">
        <base>be</base>
      </head>
      <compl xsi:type="VPPhraseSpec" FORM="PAST_PARTICIPLE">
        <head cat="VERB">
          <base>need</base>
        </head>
      </compl>
    </vp>
  </child>
</Document>
```

Figure 17. Un input dans SimpleNLG (*Refactoring is needed*), tiré du GitHub de SimpleNLG

JSreal[4] (Daoust, 2014; Daoust & Lapalme, 2015) est un réalisateur de surface pour le français écrit en JavaScript dans le but de faciliter son intégration dans les applications web. Il génère des expressions et des phrases bien formées et peut les formater en HTML pour les afficher dans un navigateur. En même temps, il peut aussi s'utiliser tout seul pour des démonstrations linguistiques. Similairement à SimpleNLG, JSreal fait la réalisation de texte à partir de spécifications d'arbres syntaxiques et à l'aide de règles syntaxiques et morphologiques et d'un dictionnaire. Ce dernier est une adaptation directe du dictionnaire de SimpleNLG-EnFr (Vaudry & Lapalme, 2013). Sa grammaire contient des règles morpho-syntaxiques qui permettent de faire l'accord entre les constituants. Il existe aussi une version bilingue anglais-français de JSreal, JSrealB (Molins & Lapalme, 2015). La Figure 18 illustre l'input permettant de réaliser la phrase *Les chats mangent une souris*.

```
S (NP (D ('le'),
       N ('chat')).n ('p'),
   VP (V ('manger'),
       NP (D ('un'),
           N ('souris'))))
```

Figure 18. Un input dans JSreal (*Les chats mangent une souris*), tiré de (Daoust & Lapalme, 2015)

Les réalisateurs profonds prennent généralement en input des structures sémantiques, plus abstraites que des structures syntaxiques. Comme nous l'avons dit, à l'intérieur de la réalisation profonde, la lexicalisation et la réalisation sont intimement liées et s'effectuent ensemble. Cela conduit au pouvoir paraphrastique de ces systèmes, c'est-à-dire qu'un même input produit souvent de nombreux outputs. Dans les réalisateurs profonds, les informations lexicales et grammaticales sont généralement encodées dans des dictionnaires et des grammaires plus complexes que ceux utilisés dans des réalisateurs de surface, afin de traiter l'interface sémantique-syntaxique.

MARQUIS (Wanner et al., 2010) est un service de génération d'information sur la qualité de l'air qui produit, en fonction de l'utilisateur, des bulletins multilingues, à partir de

[4] Une démonstration interactive de JSreal : https://observablehq.com/@lapalme/nouvelles-experiences-avec-jsrealb

données brutes. Il faut mentionner que MARQUIS est un système de GAT complet qui effectue toutes les étapes du processus de GAT. Ici, nous parlons seulement de son module de réalisation profonde (Lareau & Wanner, 2007). Comme RealPro, MARQUIS est basé sur la TST (cf. §2.1). La Figure 19 nous montre les étapes que MARQUIS accomplit pour la réalisation de texte. On peut voir que le système produit du texte à partir de structures conceptuelles. À l'intérieur de MARQUIS, chaque transformation est traitée par un module individuel qui prend en input les outputs du module précédent, jusqu'à ce que le texte soit produit. Chaque module contient des règles qui permettent le passage d'un niveau de représentation à un autre. On a besoin également d'un dictionnaire aidant à passer des unités conceptuelles aux unités sémantiques. De plus, à l'interface sémantique-syntaxique, le passage des unités sémantiques aux unités lexicales se fait à l'aide des dictionnaires sémantique et lexical ainsi que d'un dictionnaire de fonctions lexicales. Pour plus de détails, voir §2.1.

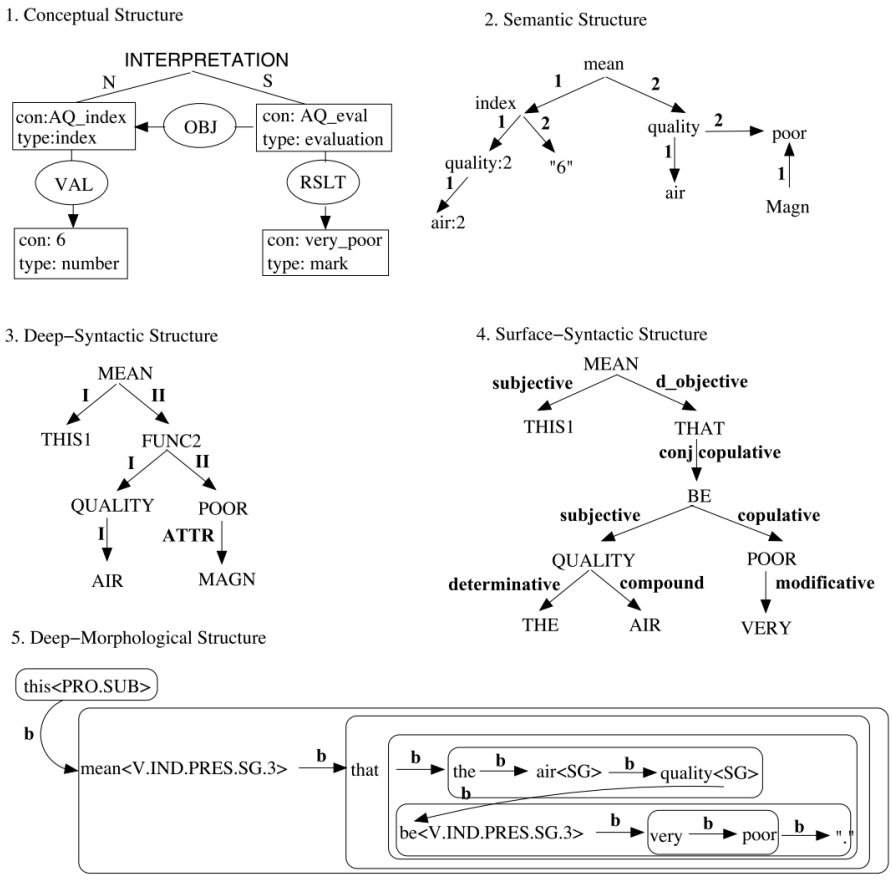


Figure 19. Déroulement de la réalisation dans MARQUIS, tiré de (Lareau & Wanner, 2007)

FORGe (Mille et al., 2017; Mille & Wanner, 2017) est un réalisateur profond à base de règles développé pour l'anglais qui est un des descendants de MARQUIS. C'est un transducteur de graphes qui génère des textes à partir d'inputs abstraits, avec des ressources lexicales. Le travail de l'adapter à d'autres langues (espagnol, allemand, français et polonais) est en développement. FORGe prend en input des représentations sémantiques sous forme de graphes orientés acycliques représentant les relations prédicats-arguments entre des sémantèmes, ce qui correspond (*Semantic Structure* dans la Figure 19). Il suit aussi le modèle théorique de la TST (cf. §2.1), puisqu'il hérite de l'architecture de MARQUIS. Il effectue les trois étapes suivantes : (i) le passage de la représentation sémantique (RSém) à la représentation syntaxique profonde (RSyntP); (ii) le passage de la RSyntP à la représentation syntaxique de surface (RSyntS); (iii) le passage de la RSyntS au texte linéarisé et morphologisé. La première étape est accomplie à travers un algorithme récursif « *top-down* ». À la deuxième étape, il s'agit d'introduire les mots fonctionnels (prépositions, auxiliaires, déterminants). Et la dernière étape finit la production du texte final en linéarisant la structure syntaxique de surface et en appliquant les règles morpho-syntaxiques aux lexèmes.

2.3 GenDR

GenDR (acronyme pour *Generic Deep Realizer*) est un réalisateur profond générique multilingue basé sur la TST (cf. §2.1) et proposé par (Lareau et al., 2018). Il s'agit d'une plate-forme pour la modélisation de l'interface sémantique-syntaxique des langues. Comme FORGe, GenDR est un descendant de MARQUIS, en héritant de l'architecture de celui-ci (cf. §2.2.2). La conception de sa grammaire est directement empruntée à MARQUIS. Néanmoins, contrairement à MARQUIS, GenDR produit seulement des structures syntaxiques de surface, qui correspondent plus ou moins à l'input des réalisateurs de surface présentés plus haut, qui complètent la réalisation jusqu'au texte. C'est-à-dire que ce réalisateur profond se concentre sur la modélisation de phénomènes langagiers profonds comme l'arborisation et la lexicalisation. Nous allons présenter ces deux tâches cruciales traitées dans GenDR à la §2.3.2, dont la stratégie est de mapper l'output de GenDR sur l'input d'un réalisateur de surface.

GenDR prend en input une représentation sémantique (à la TST) abstraite des sens et des relations prédicat-argument, et produit les structures de dépendance syntaxiques

correspondantes en anglais, français, lituanien et persan. On travaille à y ajouter d'autres langues; notre recherche consiste justement à jeter les bases d'un module spécifique pour le mandarin. Ce système fonctionne sur le transducteur de graphes MATE (Bohnet et al., 2000; Bohnet & Wanner, 2010). Dans les sections suivantes, nous présenterons l'architecture et les tâches traitées de GenDR. Nous donnerons aussi un exemple qui illustre le fonctionnement des règles et des dictionnaires dans ce système.

2.3.1 Architecture de GenDR

Dans cette section, nous présenterons les structures d'input et d'output de GenDR, ainsi que les dictionnaires et la grammaire utilisés par le système.

Le réalisateur GenDR prend en input une structure sémantique à la TST. On entre souvent cette structure sous forme textuelle dans l'éditeur de graphes de MATE, mais elle peut également être sous forme graphique. La Figure 20 et la Figure 21 montrent les deux formes d'une même structure d'input. Selon le graphe sémantique, nous pouvons voir que le prédicat est associé à ses arguments par des arcs numérotés à indiquer la position des arguments d'un même prédicat. De plus, parmi tous les sens de la structure sémantique, un doit être signalé comme le sens le plus saillant de la phrase, car il est le nœud dominant du rhème de la phrase (Mel'čuk, 2001) et qu'il sera mis en correspondance à la racine syntaxique.

Définition : Rhème

> Partie de sens 'P' d'une phrase *P* que le Locuteur présente comme information qu'il fournit. (Mel'čuk & Milićević, 2014, p. 144)

```
structure Sem debt {
  S {
    owe {
      tense=PRES
      1-> Paul {class=proper_noun}
      2-> "$500K" {class=amount}
      3-> bank {number=SG definiteness=DEF}
    }
    Paul {}
    "$500K" {}
    bank {}
    main-> owe
  }
}
```

Figure 20. Une structure sémantique sous forme textuelle

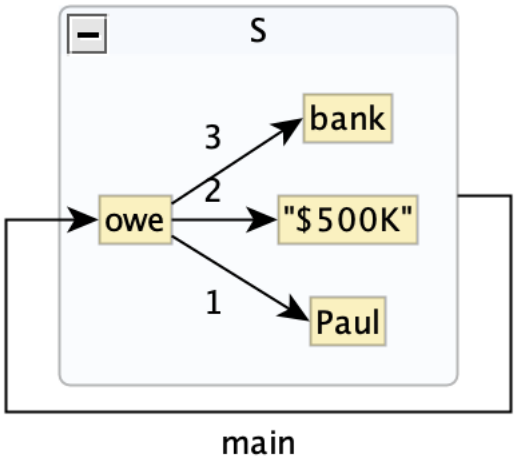


Figure 21. Une structure sémantique sous forme graphique

L'output de ce système à partir d'une structure sémantique donnée est une série de structures de dépendances syntaxiques de surface. Pour l'input à la Figure 20, GenDR peut générer six structures syntaxiques de surface (Mel'čuk, 1988), comme illustré à la Figure 22. Il s'agit ici de structures simplifiées; l'output réel est plutôt un ensemble d'arbres non ordonnés avec des lexies et des relations syntaxiques de surface. Nous montrons les vraies structures des six structures à la Figure 23. Avec l'output de GenDR, nous avons encore besoin d'un réalisateur de surface, qui le prend comme input, pour accomplir la réalisation linéarisée et fléchie.

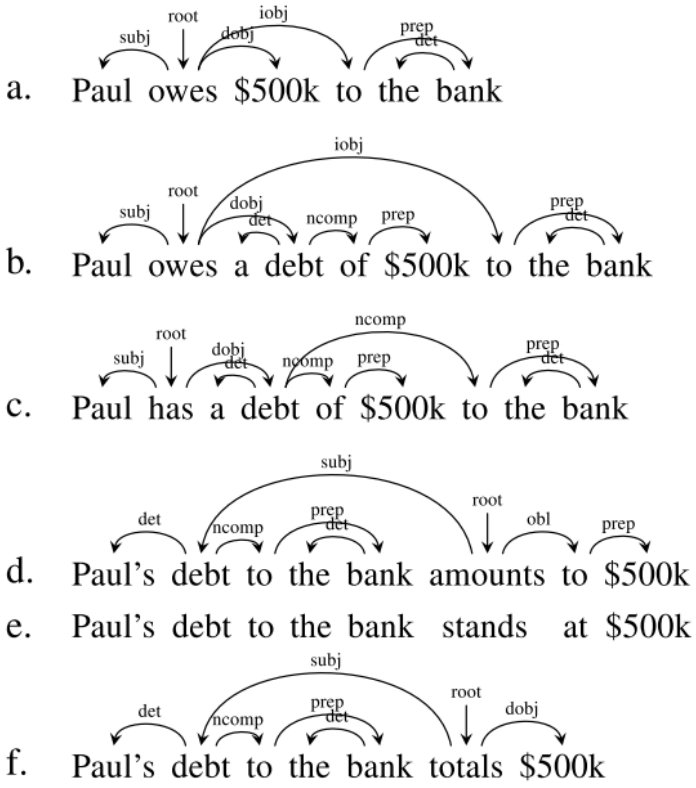
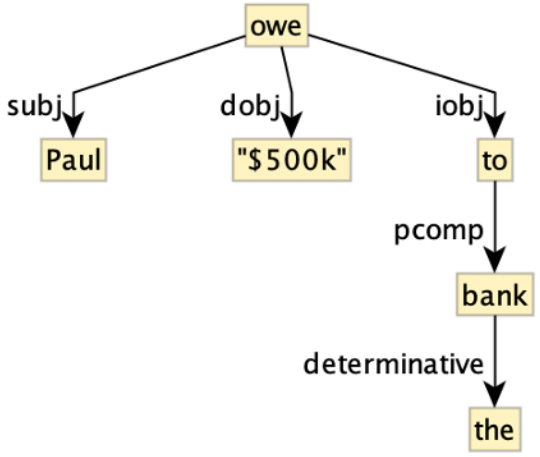
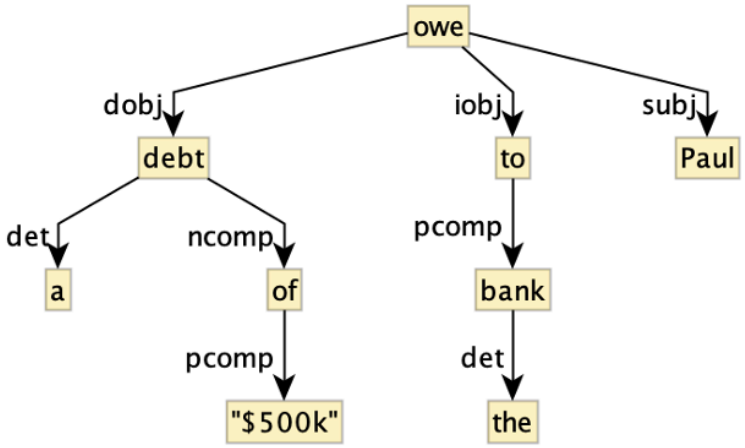


Figure 22. Six structures syntaxiques de surface simplifiées, tirées de (Lareau et al., 2018)

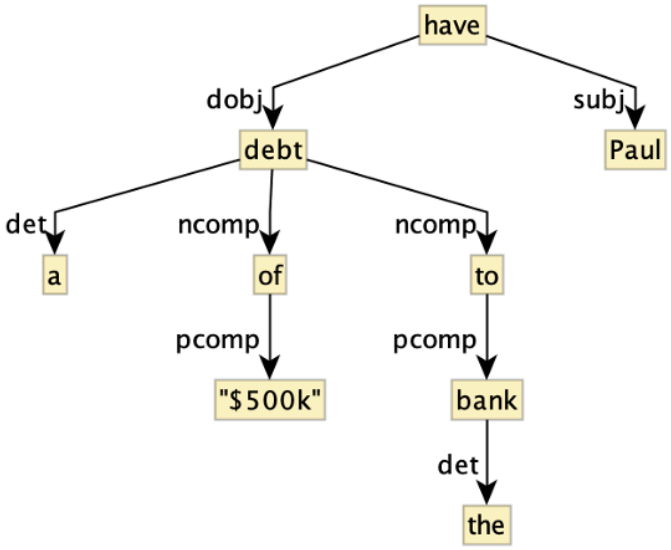
a.



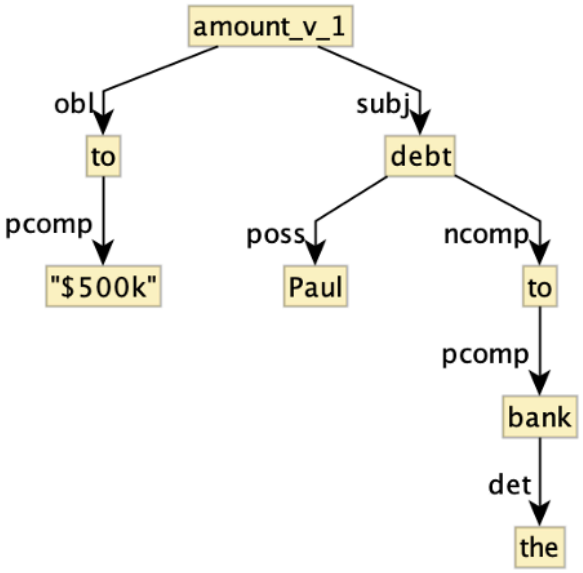
b.


owe
dobj
iobj
subj
debt
to
Paul
det
ncomp
pcomp
a
of
bank
pcomp
det
"$500k"
the

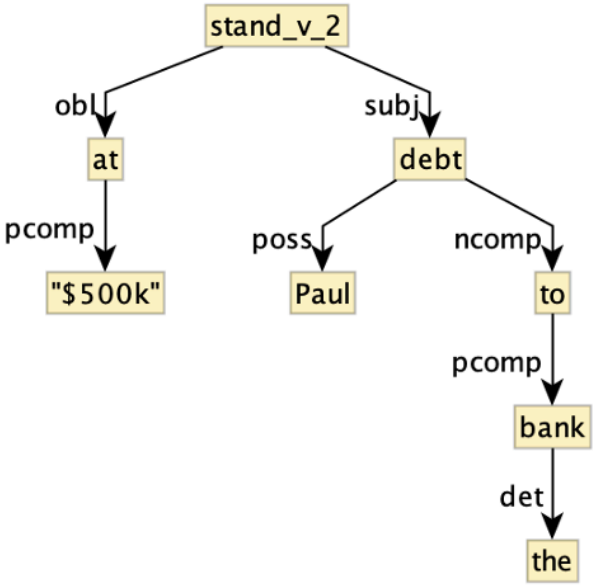
c.


have
dobj
subj
debt
Paul
det
ncomp
ncomp
a
of
to
pcomp
pcomp
"$500k"
bank
det
the

d.


amount_v_1
obl
subj
to
debt
pcomp
poss
ncomp
"$500k"
Paul
to
pcomp
bank
det
the

e.


stand_v_2
obl
subj
at
debt
pcomp
poss
ncomp
"$500k"
Paul
to
pcomp
bank
det
the

f.

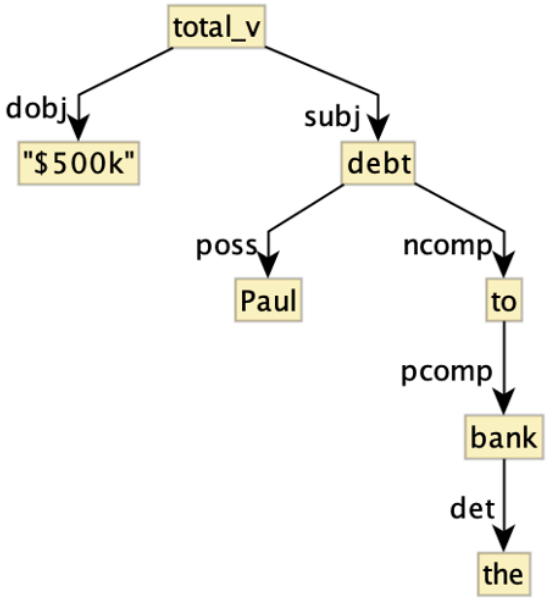


Figure 23. Six structures syntaxiques de surface produites par GenDR

Entre les niveaux de représentation d'input (représentation sémantique) et d'output (représentation syntaxique de surface) que l'on a vus, il existe également un niveau de représentation intermédiaire : la représentation syntaxique profonde (cf. §2.1). La Figure 24 illustre une structure de cette représentation intermédiaire. La réalisation de GenDR se fait en deux étapes : la transition d'une structure sémantique à une ou plusieurs structures syntaxiques profondes à l'aide des règles sémantiques-syntaxiques profonde, et la transition d'une structure syntaxique profonde à une ou plusieurs structures syntaxiques de surface à l'aide des règles syntaxiques profondes-superficielles.

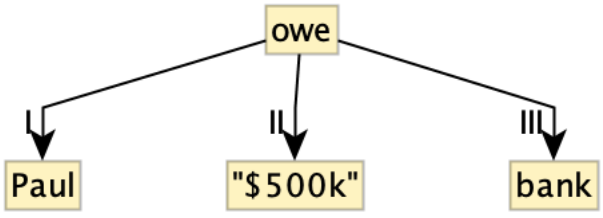


Figure 24. Une structure syntaxique profonde produite par GenDR

La réalisation de GenDR appartient aux méthodes basées sur des règles grammaticales (cf. §2.2.2), qui exigent des dictionnaires et des grammaires encodant les connaissances linguistiques d'une langue donnée. Dans le cas de GenDR, il se sert de trois dictionnaires décrivant les lexies fréquentes qui sera utilisé pour la génération de textes :

1) Le dictionnaire sémantique (*semanticon*), est un dictionnaire spécifique à chaque langue qui met en correspondance les sémantèmes avec des lexies d'une langue donnée. Cela contribue au pouvoir de paraphrasage de GenDR. La Figure 25 illustre un échantillon d'un sémantème qui correspond à deux lexies. Cette information est utilisée pendant la première étape de la réalisation (RSém-RSyntP).

```
owe { lex = owe
      lex = debt }
```

Figure 25. La description d'un sémantème dans le dictionnaire sémantique

2) Le dictionnaire lexical (*lexicon*), est un dictionnaire spécifique à chaque langue qui fournit en détail de l'information sur chaque unité lexicale d'une langue donnée. Chaque entrée doit contenir : la partie du discours profonde (`dpos`), la partie du discours de surface (`spos`), les collocations que cette lexie contrôle, et les patrons de régime (`gp`), qui incluent la diathèse (correspondance entre ses actants sémantiques et syntaxiques) et les contraintes imposées à ses actants (partie du discours, préposition, mode, définitude, etc.). GenDR utilise un mécanisme d'héritage qui permet à une entrée d'hériter des traits d'une classe abstraite (p. ex., `noun`), ce qui permet de faire des généralisations. La Figure 26 illustre un échantillon d'une lexie. On peut voir que cette description ne spécifie pas tous les traits, certains étant hérités de la classe abstraite `noun`, donnée à la Figure 27.

```
debt : noun {
  gp = {
    1 = II
    2 = I
    3 = III
    I = { dpos=Num rel=ncomp prep=of }
    II = { dpos=N rel=poss case=GEN prep=""}
    III = { dpos=N rel=ncomp prep=to }
  }
  lf = { name=Oper1 value=have }
  lf = { name=Oper13 value=owe }
  lf = { name=Func2 value=amount_v_1 }
  lf = { name=Func2 value=stand_v_2 }
  lf = { name=Func2 value=total_v }
}
```

Figure 26. La description d'une lexie dans le dictionnaire lexical

```
noun {
  dpos = N
  spos = noun
  countable = yes
}
```

Figure 27. La description de la classe abstraite `noun`

3) Le dictionnaire de fonctions lexicales (dictionnaire *LF*), est un dictionnaire qui décrit la sémantique et la syntaxe des fonctions lexicales simples et complexes (Lambrey, 2017; Lambrey & Lareau, 2015). Nous mettons ici juste un échantillon de ce dictionnaire, car notre recherche ne l'utilise pas. Pour plus d'informations sur les fonctions lexicales, voir (Kahane & Polguère, 2001; Mel'čuk, 1995; Wanner, 1996).

```
Oper1 {
  dpos = V
  gp = {  base = II
          1 = I
          I = { dpos = N
                rel = subj } } }
```

Figure 28. `Oper1` dans le dictionnaire *LF*

La grammaire de GenDR sert à traiter deux étapes de la réalisation : la transition RSém-RSyntP et la transition RSyntP-RSyntS.

À l'étape RSém-RSyntP, il s'agit de l'arborisation (transformer un graphe sémantique en arbre de dépendances syntaxique profond) et la lexicalisation profonde (mettre en correspondance les sémantèmes avec leurs lexèmes). On dispose de trois types de règles : les règles essentielles (21 règles générales empruntée à MARQUIS et 132 règles de lexicalisation) et les fonctions lexicales qui sont toutes partagées par toutes les langues, et les règles propres à chaque langue.

À l'étape RSyntP-RSyntS, il s'agit principalement de la lexicalisation de surface (ajouter des mots fonctionnels). Deux types de règles sont concernés : les règles partagées par toutes les langues et les règles spécifiques à chaque langue.

Nous allons présenter en détail les règles principales à l'intérieur de GenDR dans les prochaines sections.

2.3.2 Deux tâches cruciales traitées dans GenDR

Comme nous l'avons présenté plus haut, GenDR se consacre à la réalisation profonde (interface sémantique-syntaxique), qui concerne deux tâches cruciales à régler : l'arborisation et la lexicalisation.

Parmi les sémantèmes de la structure sémantique, un sens est marqué comme plus saillant (le nœud principal); celui-ci est mis en correspondance avec la racine de l'arbre syntaxique. À partir de ce nœud principal, l'arbre syntaxique profond est construit de manière *top-down*, en utilisant le graphe sémantique comme modèle. Cet algorithme est inspiré de (Wanner, 1992; Wanner & Bateman, 1990). L'arborisation se fait en trois étapes (Lareau et al., 2018) :

1) On construit la racine de l'arbre syntaxique en faisant référence au nœud principal de la structure sémantique, 'own'. Nous imposons des contraintes à ce nœud : pour les langues telles que l'anglais et le français, la contrainte principale est qu'il doit être un verbe à l'indicatif; cependant, cela peut être adapté pour d'autres langues telles que le chinois, qui peut avoir une racine adjectivale.

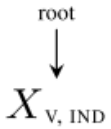


Figure 29. Première étape de l'arborisation, tirée de (Lareau et al., 2018)

2) Une fois qu'une racine a été créée et contrainte, nous cherchons une règle de lexicalisation qui est capable de satisfaire ces contraintes tout en exprimant le sens souhaité. La Figure 25 montre que le dictionnaire sémantique nous fournit deux lexicalisations concurrentes, OWE et DEBT, parmi les deux seulement le verbe OWE respecte les contraintes imposées sur le nœud syntaxique.

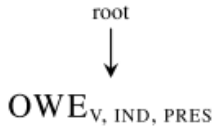


Figure 30. Deuxième étape de l'arborisation, tirée de (Lareau et al., 2018)

3) Après la lexicalisation, on commence à traiter les éléments attachés au nœud sémantique correspondant. Ses arguments, vers lesquels les flèches pointent à partir de ce nœud sémantique, doivent être réalisés comme des compléments syntaxiques. Quand la flèche pointe vers ce même nœud à partir d'un prédicat, celui-ci doit être réalisé comme un modificateur. Pour les modificateurs, on crée une dépendance avec la relation ATTR, comme illustré à la Figure 32 et à la Figure 33. S'il s'agit d'un complément, nous devons consulter le dictionnaire lexical décrivant en détail le patron de régime (`gp`), qui modélise la correspondance entre les actants sémantiques, syntaxiques profonds et syntaxiques de surface d'une unité lexicale. Les nouveaux nœuds créés à cette étape ne sont pas encore lexicalisés, mais ils se voient imposer les contraintes décrites dans le `gp` de la lexie qui les gouverne. C'est pour cela que nous avons toujours besoin d'une ressource lexicale qui détaille les contraintes. Dans le cas du verbe OWE, il y a trois arguments sémantiques. Selon la diathèse encodée dans le patron de régime (`gp = {1=I 2=II 3=III}`), le premier argument sémantique devient le premier actant syntaxique, etc. Le `gp` contraint également la partie du discours profonde des actants (cf. Figure 26).

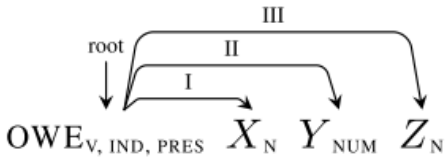


Figure 31. Troisième étape de l'arborisation, tirée de (Lareau et al., 2018)

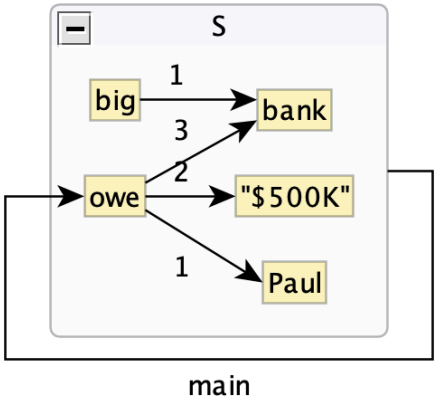


Figure 32. Une structure sémantique sous forme graphique (big bank)

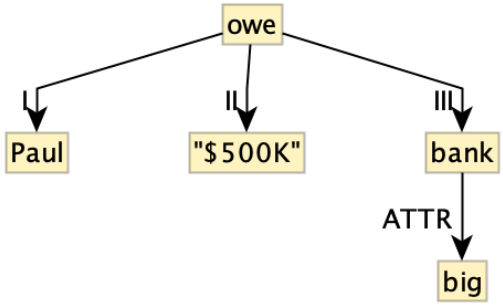


Figure 33. Une structure syntaxique profonde (ATTR)

Nous parlerons maintenant de la lexicalisation dans GenDR. Elle se fait en deux étapes. La première étape (dans l'interface RSém-RSyntP) est une lexicalisation profonde introduisant des lexies pleines et des verbes supports. Il s'agit de choisir une (ou des) unité(s) lexicale(s) profonde(s) pour exprimer une unité sémantique donnée. Ensuite, la deuxième étape (dans l'interface RSyntP-RSyntS) est une lexicalisation de surface introduisant les mots fonctionnels (prépositions, auxiliaires ou déterminants) et les lexies de surface (comme les éléments de locutions). Il s'agit de choisir les lexèmes superficiels pour exprimer les unités lexicales

profondes. Il faut mentionner que la lexicalisation s'appuie sur des ressources lexicales (trois types de dictionnaires) : un dictionnaire sémantique et un dictionnaire lexical spécifiques à chaque langue, et un dictionnaire de fonctions lexicales (cf. §2.3.1).

Concrètement, GenDR effectue six types de lexicalisation : lexicalisation simple pour les lexèmes, lexicalisation par classe pour les noms propres, les nombres, etc., lexicalisation grammaticale pour les mots fonctionnels, lexicalisation de secours pour les mots inconnus, lexicalisation par patrons pour les locutions, et lexicalisation liée pour les collocations. Nous présentons en détail les quatre premiers pour illustrer le fonctionnement de la lexicalisation dans GenDR (Lareau et al., 2018) :

1) La lexicalisation simple pour les lexèmes est la forme de lexicalisation la plus fondamentale et la plus commune. Au cours de la lexicalisation profonde, pour un nœud sémantique, nous recherchons dans le dictionnaire sémantique l'ensemble des unités lexicales qui peuvent exprimer ce sémantème donné. Parmi ces unités lexicales, on utilise seulement celles qui respectent les contraintes imposées sur le nœud syntaxique profond, en fonction de la partie du discours encodée dans le dictionnaire lexical. S'il y a plus d'une possibilité de lexicalisation pour un sémantème donné, GenDR va créer autant d'arbres profonds différents pour toutes les variantes lexicales possibles. Cela contribue au paraphrasage de GenDR.

2) La lexicalisation par classe pour les sémantèmes que nous ne voulons pas avoir une entrée dans les dictionnaires sémantique et lexical, étant donné leur grande quantité et leur comportement syntaxique prévisible : nombres, noms propres, dates, etc. Nous donnons à ces sémantèmes un trait `class`, dont la valeur pointe vers une classe correspondante qui est déjà encodée dans le dictionnaire lexical et qui fournit l'information sur la partie du discours et les traits grammaticaux. Durant la lexicalisation profonde, l'étiquette sémantique et toute caractéristique spécifiée enregistrée dans le dictionnaire lexical seront copiées sur le nœud syntaxique profond. Pendant la lexicalisation de surface, on copiera encore une fois.

3) La lexicalisation grammaticale pour les mots fonctionnels, qui ne concerne que la lexicalisation de surface. Les mots fonctionnels introduits par cette lexicalisation se

divisent en deux groupes : les uns qui expriment un sens grammatical apparaissant en attribut sur un nœud syntaxique profond donné, par exemple, les auxiliaires et les déterminants; et les autres qui sont imposés par le régime d'une lexie, par exemple, les prépositions et les complémenteurs gouvernés. Les mots fonctionnels du premier type apparaissent en tant que traits grammaticaux sur le verbe ou le nom qu'ils modifient, plutôt que comme des nœuds en syntaxe profonde. En syntaxe de surface, ils seront introduits comme un nœud supplémentaire. De plus, nous avons une règle spécifique pour chacun d'eux dans une langue, puisque ce type de mots appartiennent à des classes fermées. Quant au deuxième type de mots fonctionnels, ils ne se présentent aussi qu'en syntaxe de surface en tant que nœud supplémentaire entre une lexie qui le gouverne et son dépendant. Le système lexicalise ce type de lexies à l'aide du dictionnaire lexical, en récupérant l'étiquette lexicale dans le régime du gouverneur. Puis, l'entrée lexicale correspondante de la préposition ou du complémenteur est utilisée pour en connaître les traits spécifiques (partie du discours, etc.).

4) La lexicalisation de secours, qui sert à lexicaliser une unité sémantique ou lexicale qui n'est pas enregistrée dans les dictionnaires sémantique et lexical. En général, les ressources lexicales fournies par GenDR pour chaque langue sont assez limitées et ne renferment que les lexies les plus fréquentes. Si on rencontre la situation où le nœud sémantique n'a pas d'entrée dans le dictionnaire sémantique, la logique de GenDR est d'abord de vérifier si cette entrée se trouve sous la même forme dans le dictionnaire lexical. Si c'est le cas, on procède à une lexicalisation simple profonde à l'aide de ce lexème. Si ce mot est absent du dictionnaire lexical, le système suppose que l'étiquette de l'unité sémantique est la même que l'unité lexicale. S'il existe des contraintes de partie du discours sur le nœud, nous traitons le mot comme s'il y appartient. Ensuite, cela déclenche le régime par défaut de cette partie du discours, qui est encodé dans le dictionnaire lexical. S'il n'y a pas de contraintes sur le nœud, nous le traitons comme un nom. Les mots devinés seront signalés par un trait spécial dans la structure d'output, de sorte qu'ils puissent être filtrés pour un traitement ultérieur.

2.4 Intégration de VerbNet dans un réalisateur profond

En présentant la GAT et le réalisateur profond GenDR, nous avons montré que les ressources lexicales jouent un rôle important. Le fonctionnement de GenDR est indissociable des ressources lexicales : au cours de l'arborisation, on en récupère des contraintes pour réaliser un arbre d'output assez complet; pendant la lexicalisation, on s'en sert pour lexicaliser divers types de mots.

Quel genre de ressources lexicales nous intéresse le plus? Pour chaque langue qui s'ajoute à GenDR, on rédige les dictionnaires sémantique, lexical. Une partie importante des dictionnaires sont les données à propos des régimes de chaque unité lexicale, qui sont encodées dans le dictionnaire lexical. Précisément, ce sont les régimes de verbes que nous prenons en considération ici. Deux caractéristiques du verbe nous amènent à cette décision. Premièrement, l'imprévisibilité du régime des verbes : le verbe se différencie de la majorité des parties du discours qui possèdent des régularités à l'égard de leur régime. Deuxièmement, le rôle central que les verbes jouent dans un énoncé. Par conséquent, notre projet se concentre sur la rédaction d'un dictionnaire de régimes de verbes de la langue chinoise, qui sera ensuite intégré dans GenDR pour favoriser la création d'un module GenDR en mandarin.

Galarreta-Piquette (2018) a fait un travail remarquable d'intégration d'un dictionnaire de régimes verbaux de la langue anglaise dans GenDR. Ce travail nous a guidé sous plusieurs aspects. Dans la section qui suit, nous allons résumer les points importants de son travail : la sélection d'un dictionnaire parmi tant de candidats possibles et l'importation de VerbNet dans GenDR. De plus, nous allons présenter un dictionnaire de verbes de la langue française (VerbƎNet) comme un dictionnaire de ce type.

2.4.1 VerbNet

Contrairement à la langue chinoise, la langue anglaise est riche en ressources lexicales. Galarreta-Piquette (2018) a comparé huit dictionnaires :

- WordNet[5] (Fellbaum, 1998) est un réseau lexical qui regroupe les lexies en ensembles de synonymes en contenant des définitions et des relations sémantiques entre elles, mais il offre une description minimale des comportements syntaxiques des verbes.
- FrameNet[6] (Fillmore et al., 2003), basé sur la théorie de la sémantique des cadres (Baker et al., 1998) (cf. §1.1), fournit excellemment des données sémantiques et syntaxiques, sans classer les verbes en fonction des comportements syntaxiques.
- XTAG[7] (Paroubek et al., 1992; XTAG Research Group, 2001) est une grammaire d'arbres adjoints qui offre de l'information syntaxique en assignant à chaque unité lexicale un ensemble d'arbres décrivant les constructions syntaxiques, mais ce dictionnaire est tellement attaché à une théorie linguistique (TAG) qu'il est difficile de le convertir en TST.
- La base de données LCS[8] (Dorr, 2001) décrit les verbes en termes de leur structure conceptuelle lexicale (LCS) qui explique leur propriété syntaxique, et les regroupe en classes par le partage d'une structure LCS commune, mais cette ressources ne couvre pas autant de cadres syntaxiques que VerbNet et elle ne désambiguïse pas les différents sens des verbes.
- Comlex[9] (Grishman Macleod et Meyers, 1994) est un dictionnaire payant fournit de l'information syntaxique en décrivant les compléments possibles et les régimes pour chaque verbe, sans distinguer les sens des verbes.
- Valex[10] (Korhonen et al., 2006) est un dictionnaire de cadre sous-catégorisation (SCF) qui décrit chaque entrée lexicale avec la combinaison d'un verbe et d'un

[5] https://wordnet.princeton.edu/

[6] https://framenet.icsi.berkeley.edu/fndrupal/

[7] https://www.cis.upenn.edu/~xtag/

[8] http://users.umiacs.umd.edu/~bonnie/Demos/verbs-English.lcs

[9] https://nlp.cs.nyu.edu/comlex/

[10] https://ilexir.co.uk/valex/index.html

SCF, la syntaxe des arguments et la fréquence d'utilisation du SCF, mais son architecture n'est pas hiérarchisée.

- Le *Valence Dictionary of English* (Herbst et al., 2004) est un dictionnaire manuel de valences à petite échelle qui contient les patrons de régime des verbes les plus fréquents, il est regardé comme complément du dictionnaire de VerbNet.
- VerbNet[11] (Schuler, 2005) est construit dans le but d'offrir une ressource verbale destinée à des applications TAL. L'organisation des verbes dans ce dictionnaire se base sur la méthode de classification des verbes de Levin (1993).

C'est VerbNet qui a été retenu par Galarreta-Piquette pour l'intégration dans GenDR. Dans les paragraphes suivants, nous allons présenter d'abord la méthode de classification de Levin, l'organisation des classes verbales de VerbNet et les composantes d'une classe verbale.

2.4.1.1 Méthode de classification des verbes de Levin

Levin (1993) répartit les verbes en un nombre fini de classes verbales en fonction de comportements syntaxiques. C'est-à-dire que l'on regroupe les verbes, qui se partagent les mêmes caractéristiques sous un aspect syntaxique, dans une classe. Le partage de comportements syntaxiques communs fait que les verbes dans une même classe ont des structures d'arguments similaires. Cela ne signifie pas que ces verbes sont nécessairement synonymes. On peut voir que deux verbes synonymiques appartiennent à deux classes différentes, ou que deux verbes complètement différents peuvent appartenir à une même classe. Levin suppose en même temps que l'ensemble de cadres syntaxiques associés à une certaine classe reflète probablement des propriétés sémantiques sous-jacentes communes, qui limitent les arguments et les adjonctions autorisés. On extrait un exemple de la thèse de Schuler (2005) pour illustrer ce que Levin suppose, dans lequel on prend deux verbes en apparence synonymiques BREAK et CUT. Il semble que ces deux verbes appartiennent à une même classe, car ils contiennent tous le sens d'altérer quelque chose. Néanmoins, à travers la comparaison de leurs configurations possibles, on révèle le fait qu'ils appartiennent à deux classes différentes.

[11] http://verbs.colorado.edu/verbnet_downloads/downloads.html

(1) *Transitive Construction*

a. John broke the window.

b. John cut the bread.

(2) *Middle construction*

a. Glass breaks easily.

b. This loaf cuts easily.

(3) *Intransitive construction*

a. The window broke.

b. *The bread cut.

(4) *Conative construction*

a. *John broke at the window.

b. John valiantly cut at the frozen loaf, but his knife was too dull to make a dent in it.

On peut voir que les deux verbes sont similaires en (1) et en (2), car ils peuvent tous participer aux constructions transitive et moyenne. Cependant, on constate que, en (3) et en (4), seul BREAK autorise la construction intransitive et que seul CUT autorise la construction conative. Selon la supposition de Levin, c'est à cause de la différence de composantes sémantiques de ces deux verbes. Le verbe CUT décrit une série d'actions visant à atteindre l'objectif de séparer un objet en morceaux. Il est possible que ces actions soient effectuées sans que le résultat final soit atteint. Dans ce cas, on peut quand même percevoir que l'objet est découpé. Quant au verbe BREAK, le seul point spécifié est que le résultat de changement d'état est atteint. Sinon, aucune tentative de rupture ne peut être reconnue.

En un mot, le projet de Levin éclaire Schuler sur le regroupement des verbes en une hiérarchie de classes.

2.4.1.2 Organisation des classes verbales de VerbNet

Inspiré par la méthode de classification des verbes de Levin (1993), les verbes dans VerbNet sont regroupés en classes. VerbNet organise les classes verbales en hiérarchie; cela est inspiré de *Acquilex Lexical Knowledge Base* (Copestake, 1992) qui implémente un aspect hiérarchique à l'organisation du lexique.

Par conséquent, la sous-classe est créée : d'une part, elle hérite de toute la propriété lexicale de sa classe-mère; d'autre part, elle montre des comportements syntaxiques (constructions syntaxiques, prédicats sémantiques et restrictions sélectionnelles sur les rôles thématiques) différents de ceux de ses classes-sœurs originaires d'une même classe-mère. La Figure 34 tirée de VerbNet illustre cette organisation hiérarchique des classes verbales (CLEAR, 2005). À la Figure 34, on voit `spray-9.7` qui est le nom d'une classe verbale de VerbNet et on voit aussi `spray-9.7-1` et `spray-9.7-2` qui représentent deux sous-classes-sœurs de la classe `spray-9.7`. De même, `spray-9.7-1-1` est une sous-classe de la sous-classe `spray-9.7-1`. La logique de cette organisation hiérarchique est que la sous-classe hérite des traits de sa classe-mère et de la classe qui domine sa classe-mère et que l'une sous-classe ne partage pas ses traits avec d'autres sous-classes-sœurs.

```
<VNCLASS ID="spray-9.7">
   <SUBCLASSES>
      <VNSUBCLASS ID="spray-9.7-1">
            <VNSUBCLASS ID="spray-9.7-1-1">
      <VNSUBCLASS ID="spray-9.7-2">
         <SUBCLASSES/>
      </VNSUBCLASS>
   </SUBCLASSES>
</VNCLASS>
```

Figure 34. Organisation hiérarchique des classes verbales dans VerbNet

Il faut mentionner que la numérotation utilisée dans VerbNet est directement héritée du projet de Levin (1993). C'est une façon plus visuelle de montrer la hiérarchie d'une classe de VerbNet, comme illustré à la Figure 34. De plus, le numéro est choisi en fonction du signifié de la classe verbale, c'est-à-dire que les classes qui ont des relations sémantiques correspondantes partagent un même numéro de haut niveau (9-109). Le tableau ci-dessous est extrait de (CLEAR, 2005) qui illustre la correspondance entre le numéro et le signifié.

Class Number	Verb Type	Verb Class
9	Verbs of Putting	put-9.1 put_spatial-9.2 funnel-9.3 put_direction-9.4 pour-9.5 coil-9.6 spray-9.7 fill-9.8 butter-9.9 pocket-9.10
10	Verbs of Removing	remove-10.1 banish-10.2 clear-10.3 wipe_manner-10.4.1 wipe_inst-10.4.2 steal-10.5 cheat-10.6 pit-10.7 debone-10.8 mine-10.9 fire-10.10 resign-10.11

Tableau I. Distribution des numéros, tirée de (CLEAR, 2005)

2.4.1.3 Composantes d'une classe verbale

Chaque classe verbale comprend trois composantes :

1) Un ensemble de membres, contient la liste de verbes appartenant à une classe ou une sous-classe spécifique. L'appartenance d'un verbe est déterminée selon le projet de Levin (1993), la base de données LCS (Ayan et Dorr, 2002) et la recherche de

l'équipe de VerbNet. La Figure 35 est un exemple qui illustre les membres de la classes `give-13.1`, encodés dans VerbNet en XML.

```
<VNCLASSID="give-13.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi :noNamespaceSchemaLocation="vn_schema-3.xsd">
    <MEMBERS>
       <MEMBER name="deal"/>
       <MEMBER name="lend"/>
       <MEMBER name="loan"/>
       <MEMBER name="pass"/>
       <MEMBER name="peddle"/>
       <MEMBER name="refund"/>
       <MEMBER name="render"/>
    </MEMBERS>
```

Figure 35. Les membres de la classe verbale `give-13.1`

2) Une liste de rôles thématiques (se réfèrent à la relation sémantique entre un prédicat et ses arguments), donne de l'information sémantique. Inspiré de (Fillmore, 1968) et (Jackendoff, 1972), VerbNet a établi sa propre série de rôles thématiques qui inclut 36 rôles thématiques, dont on se sert pour identifier les arguments sélectionnés par les verbes dans chaque cadre syntaxique. Les rôles thématiques que nous utilisons dans VerbNet sont : `actor, agent, asset, attribute, beneficiary, cause, co-agent, co-patient, co-theme, destination, duration, experiencer, extent, final_time, frequency, goal, initial_location, initial_time, instrument, location, material, participant, patient, pivot, place, product, recipient, result, source, stimulus, time, theme, trajectory, topic, undergoer, value`. La Figure 36 nous montre que l'information de rôle thématique est listée dans la section <`THEMROLES`> de chaque classe verbale.

   De plus, il faut mentionner que certains rôles thématiques sont accompagnés de restrictions sélectionnelles, qui donnent plus d'informations sur la nature d'un rôle donné. Concrètement, elles imposent des contraintes aux arguments possibles. On peut voir à la Figure 36 que l'`Agent` et le `Recipient` doivent être soit un être animé, soit une organisation.

```
<THEMROLES>
   <THEMROLE type="Agent">
      <SELRESTRS logic="or">
         <SELRESTR Value="+" type="animate"/>
         <SELRESTR Value="+" type="organization"/>
      </SELRESTRS>
   </THEMROLE>
   <THEMROLE type="Theme">
      <SELRESTRS/>
   </THEMROLE>
   <THEMROLE type="Recipient">
      <SELRESTRS logic="or">
         <SELRESTR Value="+" type="animate"/>
         <SELRESTR Value="+" type="organization"/>
      </SELRESTRS>
   </THEMROLE>
</THEMROLES>
```

Figure 36. Les rôles thématiques (accompagnés de restrictions sélectionnelles)

3) Un ensemble de cadres syntaxiques, décrits dans la section <FRAMES>, fournissent une description des différentes réalisations de surface. C'est la section sur laquelle nous nous penchons le plus, car elle est très utile dans le cadre d'un réalisateur profond comme GenDR. À l'intérieur de la section <FRAMES>, il y a une sous-section <FRAME>, qui inclut une sous-section <SYNTAX>. Celle-ci fournit des informations syntaxiques (et sémantiques), en présentant un patron de régime avec un exemple de phrase. La Figure 37 illustre la présentation d'un patron de régime par lister les syntagmes selon leur ordre en surface. La Figure 38 nous montre l'organisation de la section <FRAMES>, qui est composée des constructions syntaxiques (patrons de régime), des exemples de phrase, et les rôles sémantiques mappés à des arguments syntaxiques. Les prédicats sémantiques que nous ne présentons pas ici sont également décrits dans cette section (sous la balise <SEMANTICS>), qui indiquent comment les participants sont impliqués dans l'événement.

```
<SYNTAX>
   <NP value="Agent">
      <SYNRESTRS/>
   </NP>
   <VERB/>
   <NP value="Theme">
      <SYNRESTRS/>
   </NP>
   <PREP value="to">
      <SELRESTRS/>
   </PREP>
   <NP value="Recipient">
      <SYNRESTRS/>
   </NP>
</SYNTAX>
```

Figure 37. Un exemple de la section <SYNTAX>

FRAMES	
NP V PP.DESTINATION	
EXAMPLE	"Paint sprayed onto the wall. "
SYNTAX	THEME V {{+LOC | +DIR | +DEST_CONF}} DESTINATION
SEMANTICS	MOTION(DURING(E), THEME) NOT(PREP(START(E), THEME, DESTINATION)) PREP(END(E), THEME, DESTIONATION)
NP V NP PP.DESTINATION-CONATIVE	
EXAMPLE	"The doctor squirted the eye-dropper at my eyes. "
SYNTAX	AGENT V THEME (AT) DESTINATION
SEMANTICS	MOTION(DURING(E), THEME) NOT(LOCATION(START(E), THEME, DESTINATION)) CAUSE(AGENT, E)

Figure 38. Cadres syntaxiques pour la classe `spray-9.7-1-1`, tirée de (CLEAR, 2005)

2.4.2 VerbƎNet

VerbƎNet[12] (Danlos et al., 2016) est une adaptation de VerbNet vers le français. Cette version française contient plus de 5000 emplois de verbes correspondant à environ 3000 lemmes verbaux. Les auteurs adaptent VerbNet en utilisant deux principales ressources lexicographiques françaises à large couverture pour les verbes français qui encodent le comportement syntaxique et sémantique des verbes : un dictionnaire sémantique, le LVF (*Les Verbes Français*, (Dubois & Dubois-Charlier, 1997)) qui classe environ 25000 entrées en 14 classes sémantiques, avec 54

[12] http://verbenet.inria.fr/

sous-classes syntaxico-sémantiques et 248 sous-sous-classes, et un dictionnaire syntaxique, le LG (*Lexique-Grammaire*, (Gross, 1975; Leclère, 1990)) qui répartit environ 14000 entrées en 67 tables, chacune assemblant les verbes ayant les mêmes comportements syntaxiques et éventuellement les mêmes comportements sémantiques. À l'égard de ce projet, c'est la méthode qu'ils utilisent pour créer le VerbƏNet qui nous intéresse le plus. Donc, on va présenter le développement de VerbƏNet à partir de VerbNet dans les paragraphes suivants.

La construction de VerbƏNet a été réalisée en deux étapes, tout en observant deux principes proposés dans le but d'éviter de créer une hiérarchie des classes verbaux du français et de limiter le travail aux particularités syntaxiques du français (Danlos et al., 2016). Voici les deux principes extraits de (Danlos et al., 2016) :

(i) Garder le premier niveau de la hiérarchie de VerbNet avec ses 270 classes de verbes;

(ii) Garder autant que faire se peut les informations sémantiques (attribution des rôles thématiques et décomposition sémantique des éventualités).

Comme les auteurs créent VerbƏNet en adaptant semi-automatiquement VerbNet, les deux étapes consistent à déterminer les classes verbales et les composantes de celles-ci à partir des 270 classes verbales de VerbNet. Concrètement, la première étape suit la procédure suivante (Danlos et al., 2016) :

1) Pour une certaine classe C_e de VerbNet, on trouve dans le LVF et le LG des verbes correspondant à chaque membre de C_e pour composer respectivement les classes C_{lvf} et C_{lg}. Cette correspondance repose sur la définition sémantique de C_e et sur l'invariance des rôles thématiques dans la même position syntaxique ;

2) Pour la classe C_e, dont on traduit les membres en français pour composer la liste L_{trad}, en se servant de deux dictionnaires bilingues librement disponibles (SCI-FRANEURADIC et le Wiktionnaire) ;

3) La liste des verbes français de la classe C_e, soit la classe verbale correspondante C_f, se compose des verbes à l'intersection de L_{trad}, C_{lvf} et C_{lg}.

Après la première étape, on a déterminé les verbes français qui appartiennent à chacune des 270 classes de VerbNet. Autrement dit, on a obtenu 270 classes verbales français C_f. Ensuite, les auteurs déterminent manuellement les sous-classes possibles pour assigner les verbes obtenus à la première étape à l'une de ces sous-classes et à adapter les autres composantes d'une classe verbale de VerbNet (cf. §2.4.1.3) (Danlos et al., 2016). Cette étape a besoin de définir des principes de base pour les cadres syntaxiques français qui diffèrent de ceux de l'anglais. Par exemple, les auteurs ont laissé de côté tous les cadres qui correspondent à des sous-structures avec des compléments manquants (Danlos et al., 2014; Pradet et al., 2014). Dans le VerbNet, la structure *NP V* (*Smith was incribing*) dans la classe `image_impression-25.1` pourrait être une sous-structure de la structure *NP V NP.destination* (*Smith was inscribing the rings*). Les auteurs ont expliqué la raison de cette suppression par le fait qu'ils manquent de données de corpus français.

En conclusion, VerbƎNet nous offre une méthode pour créer un dictionnaire verbal comme VerbNet d'une manière adaptative. Mais en fait, il n'y a rien qui nous laisse penser que les classes verbales du français devraient être les mêmes que les classes de l'anglais. De plus, à la deuxième étape mentionnée ci-dessus, les auteurs ont rencontré des difficultés provenant de différences basiques entre le français et l'anglais. De notre point de vue, cette méthode n'est pas appropriée pour créer le dictionnaire tel que VerbNet pour d'autres langues qui sont largement différentes de l'anglais, par exemple le mandarin.

2.4.3 Importation de VerbNet dans GenDR

Galarreta-Piquette (2018) a importé VerbNet dans GenDR d'une façon automatique à l'aide de scripts Python qui extraient des fichiers XML. Le cœur de l'importation de VerbNet dans GenDR réside dans la création des dictionnaires utilisés dans GenDR : *semanticon* (dictionnaire sémantique), *lexicon* (dictionnaire lexical) et *gpcon* (dictionnaire des régimes). Parmi les trois dictionnaires, seul le *semanticon* que l'auteur utilise est sensiblement le même que celui utilisé dans la version initiale de GenDR. Cette importation consiste à préparer le nouveau *lexicon* de GenDR et à créer le *gpcon*.

Au cours de l'importation, l'auteur a créé trois dictionnaires temporaires qui sont en effet les outputs écrits dans des fichiers *.dict* après la manipulation et l'extraction des données de

VerbNet, dans le but de former les deux dictionnaires employés dans GenDR (*lexicon* et *gpcon*) (Galarreta-Piquette, 2018) :

1) `lexicon.dict` est un fichier qui enregistre le résultat de l'extraction de l'architecture de VerbNet, y compris la hiérarchie de VerbNet et les classes verbales. La Figure 39 et la Figure 40 illustrent les données que le script prend en input et ce qu'il génère en output. Il faut mentionner que l'auteur reprend le mécanisme d'héritage servant à organiser le dictionnaire lexical de GenDR (cf. §2.3.1) pour importer des classes verbales de VerbNet, puisque ce mécanisme ressemble à l'architecture de VerbNet (cf. §2.4.1.2) qui consiste à organiser les classes verbales en hiérarchie : la classe-fille hérite des propriétés d'une classe qui la domine. De plus, l'auteur attache la classe abstraite à chaque classe-mère, pour que les traits de partie du discours puissent être transmis aux verbes appartenant à cette classe-mère.

```
<FRAMES>
      <FRAME>
          <DESCRIPTION    descriptionNumber="0.2"    primary="NP    V    NP"
secondary="Basic Transitive"/>
          <EXAMPLES>
              <EXAMPLE>Cotton absorbs water.</EXAMPLE>
          </EXAMPLES>
          <SYNTAX>
              <NP value="Goal">
                  <SYNRESTRS/>
              </NP>
              <VERB/>
              <NP value="Theme">
                  <SYNRESTRS/>
              </NP>
          </SYNTAX>
      <FRAME>
       …
```

Figure 39. Input du script : cadres syntaxiques imbriqués dans les fichiers VerbNet, tirée de (Galarreta-Piquette, 2018)

```
lexicon {
"absorb-39.8": verb {
  gp = { id=NP_V_NP dia=x } // Cotton absorbs water.
  gp = { id=NP_V_NP_PP_from_source dia=x } // Cattle take in nutrients from
their feed.
}
```

Figure 40. Output du script : propriétés de la classe verbale `absorb-39.8`, tirée de (Galarreta-Piquette, 2018)

En récupérant l'identifiant de la classe verbale, les identifiants des patrons de régimes de celle-ci et une phrase exemple à l'aide de scripts Python, l'auteur compte créer un dictionnaire de classes verbales. De plus, les prépositions régies par chaque patron de régime sont ajoutées à l'identifiant de celui-ci, pour que les classes verbales puissent sélectionner les bonnes prépositions.

2) `members.dict` est un fichier qui enregistre le résultat de l'importation de plus de 6000 verbes décrits dans VerbNet, comme illustré à la Figure 42. Ce qui est important au cours de cette importation, c'est de désambiguïser la forme des verbes des différentes acceptions d'un même vocable. L'auteur récupère les verbes, en les attachant à sa classe verbale correspondante de VerbNet. Puis, les verbes sont classés par ordre alphabétique.

```
<VNCLASS ID="absorb-39.8" >
    <MEMBERS>
        <MEMBER name="absorb">
        <MEMBER name="ingest" >
        <MEMBER name="take_in">
    </MEMBERS>
```

Figure 41. Input du script : verbes correspondant aux membres d'une classe verbale, tirée de (Galarreta-Piquette, 2018)

```
abound : "swarm-47.5.1-2-1"
abrade : "other_cos-45.4"
abridge : "other_cos-45.4"
absolve : "free-80-1"
absorb : "absorb-39.8"
```

Figure 42. Output du script : lexèmes pointant vers une classe verbale, tirée de (Galarreta-Piquette, 2018)

3) `gpcon.dict` est un fichier qui liste tous les patrons de régime (`gp`) sous la forme attendue par GenDR, comme illustré à la Figure 43. L'auteur extrait les propriétés nécessaires pour construire le *gpcon* des identifiants des `gp` (p. ex., `NP_asset_V_NP_PP_from_out_of`) et des phrases exemples, prélevés lors de la création du fichier `lexicon.dict` : la partie du discours de chaque actant syntaxique, l'ordre des actants syntaxiques en RSyntP, les prépositions régies par le verbe pour un actant donné et les relations de surface (sujet, objet direct, etc.) de chaque actant syntaxique.

```
gpcon {
NP_agent_V {
   I={rel=subjective dpos=N}
}
NP_agent_V_NP {
   I={rel=subjective dpos=N}
   II={rel=dir_objective dpos=N}
}
NP_asset_V_NP_PP_from_out_of {
   I={rel=subjective dpos=N}
   II={rel=dir_objective dpos=N}
   III={rel=oblique dpos=N prep=from}
   III={rel=oblique dpos=N prep="out of"}
}
NP_attribute_V {
   I={rel=subjective dpos=N}
}
...
```

Figure 43. Extrait du *gpcon* : liste des patrons de régime, tirée de (Galarreta-Piquette, 2018)

Après avoir défini des objets (p. ex., `subj = 'rel=subjective dpos=N', dir_N = 'rel=dir_objective dpos=N'`) représentant des propriétés syntaxiques qui seraient associés à un actant syntaxique et les ajoutés à la description de chaque `gp` (p. ex., `{'NP_agent_V_ NP': [subj, dir_N],}`), l'auteur obtient finalement le patron de régime (voir la Figure 43) qu'exige GenDR, à l'aide de scripts Python.

En conclusion, l'importation de VerbNet dans GenDR consiste à extraire les informations de VerbNet et à les adapter pour GenDR. Galarreta-Piquette (2018) nous a offert plusieurs méthodes utiles qui méritent d'être une référence pour l'importation de notre travail dans GenDR ou d'autres réalisateurs.

Chapitre 3 Méthodologie de la recherche

3.1 Choix de la ressource lexicale

Au tout début, nous avions l'intention de créer un module GenDR spécifique pour le mandarin, et il nous fallait trouver des ressources lexicales appropriées pour ce faire. Nous avons présenté le VerbNet (cf. §2.4.1), qui favorise largement la création du module anglais. Néanmoins, nous n'avons pas trouvé de dictionnaire verbal équivalent à VerbNet en mandarin. Étant donné ce fait, nous avons décidé de procéder à la recherche présente : créer un dictionnaire de régimes verbaux en mandarin, en tant que base sur laquelle nous pourrions ensuite créer un module pour le mandarin.

Parmi toutes les ressources lexicales que nous avons consultées, il y en a une qui a attiré notre attention, *Mandarin VerbNet* (Liu & Chiang, 2008), puisqu'elle est assez proche du VerbNet anglais. Néanmoins, même si cette ressource s'appelle «VerbNet», elle n'est pas vraiment un dictionnaire verbal de type VerbNet. En fait, *Mandarin VerbNet* se base principalement sur la théorie de la sémantique des cadres (Baker et al., 1998). Dans ce dictionnaire, les verbes sont classés dans différents cadres en fonction des rôles sémantiques de leurs arguments et de leurs patrons de construction. En bref, nous ne pouvons pas obtenir directement les propriétés des verbes qui correspondent à notre besoin à partir de ce dictionnaire (voir §1.1). Malgré ce fait, nous avons décidé de nous en servir comme base pour la création d'un dictionnaire de régimes verbaux en mandarin. Dans les paragraphes suivants, nous allons présenter quelques ressources alternatives, puis expliquer la raison pour laquelle nous avons finalement choisi le *Mandarin VerbNet*.

Dans notre cas, la ressource que nous cherchions n'était pas simplement un corpus assez grand et riche en mandarin. Nous avions deux exigences essentielles à satisfaire :

1) La ressource peut être téléchargée et traitée par des scripts Python, ce qui nous permet d'extraire les informations nécessaires;

2) La ressource doit avoir une grande couverture pour offrir assez de données pour créer un dictionnaire verbal pour la génération de texte.

En fonction des exigences mentionnées ci-dessus, nous avons éliminé plusieurs corpus (p. ex., celui du BLCU Corpus Center[13], ou celui du Center for Chinese Linguistics PKU[14]). Nous avons examiné plus en détail deux ressources : le corpus annoté en dépendances UD_Chinese-GSD[15] offert par le projet Universal Dependencies[16] (UD) (Nivre et al., 2016) et le corpus *Mandarin VerbNet*. Il faut noter bien que nous comptons seulement les patrons de construction et les phrases d'exemple du dictionnaire *Mandarin VerbNet* comme **le corpus Mandarin VerbNet** dont nous nous servons dans cette recherche.

Le premier est un corpus de phrases annotées avec des étiquettes de partie du discours et des relations syntaxiques de type Universal Dependencies. Ce corpus contient 4 997 phrases et 123 291 tokens. Le schéma d'annotation est basé sur une évolution des dépendances (universelles) de Stanford (de Marneffe et al., 2006, 2014; de Marneffe & Manning, 2008), des étiquettes de partie du discours universelles de Google (Petrov et al., 2012), et une approche universelle de conversion entre des séries d'étiquettes morphosyntaxiques de plusieurs langues (Zeman, 2008). Nous avons utilisé l'outil en ligne Grew-match[17], qui permet à l'utilisateur de chercher un patron donné dans un corpus UD, ce qui nous a servi à trouver des phrases contenant des prépositions liées aux verbes pour les analyser.

En ce qui concerne le deuxième corpus, Prof. Liu de la City University of Hong Kong (responsable du projet *Mandarin VerbNet*) a gentiment accepté de nous donner accès aux fichiers de la base de données *Mandarin VerbNet*[18], qui est normalement seulement consultable en ligne. Dans ces fichiers, chaque entrée se compose de trois parties : écriture en pinyin du

[13] http://bcc.blcu.edu.cn/

[14] http://ccl.pku.edu.cn:8080/ccl_corpus/

[15] https://universaldependencies.org/treebanks/zh_gsd/index.html
https://github.com/UniversalDependencies/UD_Chinese-GSD

[16] https://universaldependencies.org/

[17] http://match.grew.fr/

[18] http://mega.lt.cityu.edu.hk/~yufechen/#/

verbe, fréquence, et patrons de construction composés du *Basic Frame* en jeu et certains *frame elements* (avec plusieurs phrases illustratives). Nous prenons les patrons de construction et les phrases illustratives comme le corpus. Étant dans un format lisible par une machine, ce corpus nous permet d'extraire les informations dont nous avons besoin avec des scripts Python. Ce qui est spécifique, c'est que ce corpus est une ressource qui ne concerne que les verbes. Toutefois, cette base de données ne donne que des informations limitées pour chaque verbe. Comme ce dictionnaire est principalement basé sur la théorie de la Sémantique des cadres (Baker et al., 1998), dont l'idée centrale est que le sens des lexies doit être décrit en termes de cadres sémantiques qui décrivent les interactions sémantiques entre la lexie décrite et les participants de la situation dénotée (appelé *frame elements*), presque 500 verbes sont répartis en 16 *Archi Frames*, avec 104 *Basic Frames*.

Finalement, nous avons décidé d'exploiter le corpus *Mandarin VerbNet* plutôt que le corpus UD_Chinese-GSD, en tant que ressource lexicale de notre projet. Les raisons sont énumérées ci-dessous :

1) Ce corpus nous offre seulement de l'information sur des verbes, qui est la partie du discours qui nous intéresse. Ainsi, nous n'avons pas besoin d'isoler les verbes dans une ressource moins spécifique.

2) Ce corpus nous offre assez de données pour prélever et analyser les propriétés des verbes (p. ex., préposition régies par le verbe, patron de régime). Nous avons mentionné que plusieurs exemples illustrent chaque patron de construction. Or, ces exemples sont annotés avec *Basic Frame* et *frame elements*, ce qui favorise le prélèvement des informations qui nous intéressent.

Jusqu'ici, nous avons présenté le choix de la ressource lexicale et en même temps le corpus choisi et sa version original *Mandarin VerbNet*. Dans les paragraphes suivants, nous allons présenter de façon générale comment nous avons exploité ce corpus.

3.2 Corpus complémentaires à préparer

Dans le but de créer un dictionnaire de régimes verbaux en mandarin, nous avons extrait les adpositions de la ressource lexicale choisie. Pour ce faire, nous devions d'abord élaborer une

liste d'adpositions de référence (cf. §3.3.2). L'élaboration de cette liste est basée sur deux ressources : 《现代汉语词典》(*Dictionnaire du Mandarin Moderne*)[19] et le treebank UD_Chinese-GSD (cf. §3.1). Le premier nous offre un tableau d'adpositions. Le Tableau II est une version reproduite et traduite manuellement. En plus des adpositions listées dans 《现代汉语词典》, nous avons extrait les adpositions du corpus UD_Chinese-GSD en utilisant Grew-match[20] : nous cherchons une partie du discours donnée (ADP) en entrant ce qui illustré à la Figure 44, puis nous obtenons un tableau qui contient toutes les adpositions trouvées dans ce corpus. En combinant les deux ensembles d'adpositions, nous avons créé une liste d'adpositions préliminaire.

Il reste encore trois étapes dans la préparation des ressources complémentaires, qui sont tous faits à la main :

1) Nettoyer la liste préliminaire pour supprimer les doublons et le bruit;
2) Ajouter les graphies alternatives pour chaque mot. Nous avons trouvé que les deux formes (simplifiée et traditionnelle) se confondent dans beaucoup de corpus chinois. De plus, les deux formes traditionnelles (de Taïwan et de Hong Kong) se confondent souvent. Ainsi, si une adposition a trois formes (forme simplifiée, forme traditionnelle de Taïwan et forme traditionnelle de Hong Kong), elle va apparaître trois fois sous différentes formes dans la liste d'adpositions finale. Le Tableau III illustre les différentes graphies pour les adpositions dans la liste.
3) Grouper les adpositions en deux listes : prépositions et postpositions, puisque les prépositions et les postpositions seront détectées séparément au cours de l'extraction des adpositions (cf. §3.3.2).

---

[19] C'est le premier dictionnaire de mandarin standard compilé par le bureau d'édition du dictionnaire de l'Institut des langues de l'Académie chinoise des sciences sociales, publié par la Presse Commerciale en 1978 et réédité plusieurs fois. Sa dernière version (7e édition) a été publiée en 2016, qui comprend 70 000 entrées (cf. https://zh.wikipedia.org/wiki/现代汉语词典 [consulté le 27 oct. 2020]).

[20] Grew-match est une application web en ligne pour chercher des modèles de graphes dans les treebanks. (cf. http://match.grew.fr/)

Adpositions dans le *Dictionnaire du Mandarin Moderne* (83 au total)	
55 monosyllabiques	28 dissyllabiques
àn 按 běn 本 bǎ 把 bèi 被 bēn 奔 bǐ 比 bìng 并 cháo 朝 chī 吃 chōng 冲 chú 除 cóng 从 dǐng 顶 gǎn 赶 guǎn 管 dǎ 打 duì 对 gěi 给 hé 和 jiù 就 ná 拿 qí 齐 qǐ 起 ràng 让 shùn 顺 tì 替 gēn 跟 guī 归 jiāng 将 jiǎng 讲 jiào 叫 jiè 借 kě 可 lián 连 lín 临 jìn 尽 tóu 头 rú 如 tóng 同 wéi 为 wèn 问 xiàng 向 yú 于 yuán 缘 yán 沿 yǐ 以 yǐ 迤 yīn 因 yóu 由 yǔ 与 zài 在 zhǎng 掌 zhào 照 zhǔn 准 zì 自	ànzhào 按照 bǐjiào 比较 chúfēi 除非 chúkāi 除开 chúle 除了 chúqù 除去 chúquè 除却 guānyú 关于 cóngdǎ 从打 cuòfēi 错非 dǎcóng 打从 duìyú 对于 gēnjù 根据 jīyú 基于 jiànyú 鉴于 jízhì 及至 jīngyóu 经由 tōngguò 通过 wèile 为了 wèizhe 为着 yīcóng 一从 yījù 依据 yīnwéi 因为 yóudǎ 由打 yóuyú 由于 zìdǎ 自打 zìcóng 自从 zuòwéi 作为

Tableau II. La version traduite du tableau d'adpositions dans 《现代汉语词典》

```
pattern { N [upos="ADP"] }
```

Figure 44. Patron de recherche entré dans Grew-match pour chercher une partie du discours

Graphie dans la liste	Graphie ajoutée dans la liste
wèi 为 (forme simplifiée) ‘pour’	為 (forme traditionnelle de Taïwan) 爲 (forme traditionnelle de Hong Kong)
lǐ 裡 (forme traditionnelle de Taïwan) ‘dans’	里 (forme simplifiée) 裏 (forme traditionnelle de Hong Kong)
duì 对 (forme simplifiée) ‘à’	對 (forme traditionnelle uniforme)
cóng 從 (forme traditionnelle uniforme) ‘de’	从 (forme simplifiée)
yǐ 以 (forme uniforme) ‘avec’	N/A

Tableau III. Addition de formes alternatives des adpositions

Après toutes les étapes, nous avons obtenu au total 85 adpositions, parmi lesquelles 42 prépositions et 43 postpositions. En considérant les graphies alternatives, nous avons 83 formes dans la liste des prépositions et 68 formes dans la liste des postpositions.

3.3 Extraction de l’information utile de la ressource choisie

3.3.1 Normalisation

La base de données *Mandarin VerbNet* à laquelle nous avons accès se compose de 420 fichiers (chacun contenant une entrée du dictionnaire). Normalement, chaque entrée contient trois parties : pinyin, fréquence et patrons de construction composés de *Basic Frame* qui représente le verbe en jeu et certains *frame elements* (chacun avec plusieurs phrases d’exemple), comme illustré à la Figure 45. À l’égard des phrases d’exemple, le *Basic Frame* en jeu (en majuscule) et chaque *frame element* (dont la première lettre en majuscule) sont mis entre crochets et sont annotés avec le nom de *frame element* correspondant, qui est déjà mentionné

dans le patron de construction. Dans ce corpus brut, nous avons trouvé quelques problèmes qui nous ont poussé à faire une normalisation avant l'extraction.

```
## pin yin: shuō
## frequency: 2643990
%%PATTERN: [Speaker]-[SAY]-[Message](98/151)
[媽媽/Speaker][說/SAY][我小時候很調皮/Message]。
先前[你/Speaker]又沒[說/SAY][要給我拔針/Message]。
......
%%PATTERN: [Speaker]-[Addressee]-[SAY]-[Message](6/151)
[章文濤先生/Speaker]在車上笑著對[我/Addressee][說/SAY]，[雖然全車除了我之外都是山西人，但這次旅行的嚮導應該是我，原因只在於我讀過一些史料/Message]。
[她/Speaker]回去之後，卻也沒有跟[計爺爺/Addressee]說起，只[說/SAY][在大漠中迷了路，越走越遠，幸好遇到一隊駱駝隊，才不致渴死在沙漠之中/Message]。
......
```

Figure 45. L'entrée shuō 說 'dire' du corpus brut, extrait de la base de données *Mandarin VerbNet*

Voici des problèmes trouvés dans ce corpus brut :

1) Comme le patron de construction que nous voyons à la Figure 45, le *Basic Frame* en majuscule représente le verbe en jeu. Toutefois, l'auteur a ajouté des patrons de construction à propos de ce mot utilisé comme nom dans la même entrée.

```
## pinyin: chǎo jià
## freque ncy: 1161
%%PATTERN: [Intl_1]-[Intl_2]-[QUARREL](63/206)
[他/Intl_1]與[於秀芝/Intl_2]昨天晚上就開始[吵架/QUARREL]
......
%%PATTERN: [Intl_1]-[Intl_2]-[*V_nom]-[*QUARREL+nom](3/206)
[爺爺/Intl_1]與[奶奶/Intl_2]又[發生/*V_nom][吵架/*QUARREL+nom]
......
```

Figure 46. L'entrée chǎojià 吵架 'se quereller' du corpus brut, extrait de la base de données *Mandarin VerbNet*

À la Figure 46, on peut voir que le *Basic Frame* dans le deuxième patron n'est pas le même que celui dans le premier : «+nom» est ajouté entre crochets. Dans ce cas, l'entrée en jeu est utilisée comme nom au lieu d'un verbe. Comme le dictionnaire que

nous sommes en train d'établir ne concerne que les verbes, nous devons retirer ce genre de patron du corpus. Cette étape se réalise au moyen de scripts Python.

2) Dans le but de créer un dictionnaire de régimes verbaux en mandarin, on doit mettre en lumière les propriétés des verbes (p. ex., préposition régie par le verbe, patron de régime). L'étape d'extraction consiste donc à détecter les adpositions se trouvant avant ou après chaque *frame element* (en crochet avec la première lettre en majuscule). Néanmoins, dans certains cas, les adpostions possibles sont incluses dans le *frame element*, soit entre les crochets. Dans ce cas, une autre tâche de la normalisation consiste à sortir les adpositions possibles des crochets de *frame elements*. La Figure 47 illustre comment on normalise ce genre de cas. Cette étape est faite par un script Python.

[**prep**ABCD/*Frame element*] ⟹ **prep**[ABCD/*Frame element*]

[ABCD**postp**/*Frame element*] ⟹ [ABCD/*Frame element*]**postp**

Figure 47. Normalisation de l'annotation des adpositions pour un *frame element*

3) Dans certains cas, les adpositions possibles sont mises entre crochets en tant que *frame element*. Il nous faut effacer les crochets pour que nous puissions bien détecter les adpositions possibles à l'étape de l'extraction. Cette étape illustrée à la Figure 48 se réalise à l'aide d'un script Python.

[**prep**/*Tar_mkr] ⟹ **prep**

Figure 48. Élimination des crochets pour les adpostions

3.3.2 Extraction des adpositions

Une fois le corpus normalisé, nous pouvons en extraire l'information utile.

En premier lieu, on doit savoir de quel genre d'information on a besoin afin de créer un dictionnaire de régimes verbaux. Pour chaque entrée, la partie de patrons de construction est la plus importante : on se sert des phrases d'exemple pour trouver les adpositions possibles et calculer la fréquence d'apparition de celles-ci pour chaque *frame elements*.

À l'aide de scripts Python, nous avons extrait la partie de patrons de construction et à le ranger comme illustré à la Figure 49. Avant tout, il faut présenter quelques notions élémentaires de Python. La Figure 50 nous donne la formule d'un dictionnaire en Python, qui est capable de rassembler des éléments en tant que clés. À chaque clé est associée une valeur. La liste en Python illustré à la Figure 51 est un type de données (mis entre crochets) qui collecte des éléments séparés par des virgules. Dans le dictionnaire en Python que nous avons fait à la Figure 49, chaque patron de construction est une clé à laquelle correspond une valeur qui est une liste en Python des phrases d'exemple annotées.

```
{'[Speaker]-[EXPRESS]-[Message]' :
[
'史達林元帥重視中國方面的意見，就是[外蒙人民/Speaker]投票[表示/EXPRESS][願意獨立/Message]之後，再宣佈外蒙獨立。',
'但由於[毛澤東/Speaker]在一九五六年四月的中央政治局會議上曾[表示/EXPRESS][建設應加速/Message]，主張追加基本建設的預算投資。',
'......',
'......'
]

'[Speaker]-[Topic]-[EXPRESS]-[Message_Description]':
[
'[Ｂ型和人馬座/Speaker]的性格有很多共通點自由意志性也是其中之一。好奇心強，對[什麼事/Topic]都[表示/EXPRESS][興趣/Message_Description]，對事情不執著的直爽之氣質等，也是共通的。',
'[Ｂ型—雙魚座/Speaker]的好奇心也很旺盛，對[身邊發生的種種事情/Topic]都[表示/EXPRESS][關心/Message_Description]。',
'......',
'......'
]
......
}
```

Figure 49. Patrons de construction et phrases d'exemple extraits

```
dict = {clé1 : valeur1, clé2 : valeur2, clé3 : valeur3 }
```

Figure 50. Formule d'un dictionnaire en Python

```
jour = ["lundi", "mardi", "mercredi", 1800, 20.357, "jeudi", "vendredi"]
```

Figure 51. Un exemple de liste en Python

Ensuite, on prélève les caractères à droite et à gauche de chaque *frame element* et en même temps on se sert des listes d'adpositions préparées pour détecter s'il y a une adposition dans les mots prélevés. Comme la longueur maximum d'adposition enregistrée dans la liste d'adpositions préparées est deux caractères, on prélève seulement deux caractères à droite et à gauche de chaque *frame element*. La Figure 52 illustre un exemple de ce prélèvement. Après le prélèvement, on vérifie s'il y a une adpositon dans les caractères prélevés. Si c'est le cas, l'adposition détectée, le lemme et le *frame element* correspondants sont extraits. Avec ces informations extraites, nous calculons ensuite la fréquence d'apparition pour chaque adposition pour chaque *frame element*. Tout cela se réalise à l'aide de scripts Python.

[媽媽/Speaker]在公交車上笑著對[我/Addressee][說/SAY]，[我小時候很調皮/Message]。

māmā zài gōngjiāochēshàng xiàozhù duì wǒ shuō wǒ xiǎoshíhòu hěn tiáopí
媽媽 在 公交車上 笑著 對 我 說， 我 小時候 很 調皮。
mère être bus-sur souriant pour moi dire moi dans l'enfance très fripon
'Dans le bus, ma mère m'a dit avec un sourire que j'étais très fripon quand j'étais petit.'

Frame Element	**Deux caractères prélevés à gauche**	**Deux caractères prélevés à droite**
Speaker	—	在公
Addressee	著對	—
Message	—	—

Figure 52. Un exemple de prélèvement

À la fin, notre script génère un tableau comme illustré au Tableau IV. Chaque ligne de ce tableau doit se lire comme suit : à propos d'un certain *frame element*, on a trouvé une préposition ou une postposition, dont la fréquence d'apparition est la valeur correspondante à la colonne «Fréquence relative». À l'égard d'un certain *frame element*, le calcul de la fréquence relative d'une adposition est que nous divisons sa nombre d'apparition par la somme du nombre d'apparition de tous les cas. Une barre «—» signifie que ce *frame element* peut être exprimé sans adposition. La fréquence d'apparition de ce cas est aussi calculée. Pour chaque *frame element* d'un lemme, la somme de la fréquence d'apparition de tous les cas doit être 1.

Verbe	***Frame Element***	**Type**	**Adposition**	**Fréquence relative**
說 ‘dire’	Addressee	—	—	0.556
說	Addressee	préposition	和 ‘avec’	0.028
說	Addressee	préposition	跟 ‘avec’	0.194
說	Addressee	préposition	向 ‘à’	0.028
說	Addressee	préposition	給 ‘à’	0.056
說	Addressee	préposition	對 ‘à’	0.139
說	Message	—	—	1.000
說	Topic	—	—	0.500
說	Topic	préposition	關於 ‘à propos de’	0.500
…	…	…	…	…

Tableau IV. Tableau généré à la fin de l’extraction

Jusqu’ici, le traitement du corpus brut finit par extraire les adpositions à partir des phrases d’exemple et obtenir leur fréquence relative. Ces informations nous aide à créer le dictionnaire verbal. Dans le paragraphe suivant, nous allons présenter comment nous traitons ces informations utiles pour créer un dictionnaire désiré.

Chapitre 4 Dictionnaire de régimes des verbes

4.1 Adpositions régies

Les adpositions recueillies par le procédé expliqué plus haut (cf. §3.3) sont toutes celles qui sont trouvées immédiatement à droite ou à gauche d'un *frame element* dans le corpus. Cela ne signifie pas que ce sont toutes des adpositions régies par le verbe en jeu. Ici, nous précisons deux critères nécessaires pour qu'une adposition ainsi extraite soit considérée comme régie :

1) Cette adposition doit être contrôlée syntaxiquement par le verbe;
2) Il doit y avoir une association forte entre le verbe et l'adposition. C'est-à-dire que si on change le verbe, l'adposition ne peut pas toujours être conservée. Nous prenons ici deux phrases en français pour illustrer ce critère. Nous pouvons voir que le verbe *rentrer* s'emploie avec la préposition *de*. Si on change le verbe, la conservation de la préposition *de* pourrait entraîner une erreur grammaticale. À l'égard du verbe *rentrer*, *de* est une adposition régie.

   Luc rentre de Québec.

   **Luc va de Québec.*

Dans notre recherche, nous nous consacrons à trouver les adpositions et à les vérifier si les adpositions trouvées sont régies par le verbe en jeu à l'aide des deux critères mentionnés. Ce sont la théorie sur laquelle le corpus Mandarin VerbNet se base et le format d'annotation du corpus qui nous font décider de trouver les adpositions autours des *frame elements*. Cela ne signifie pas que l'adposition dépend juste du *frame element*. Le travail que nous avons fait est insuffisant pour discuter la question : De quoi dépend l'adposition? Il faut plus d'analyse sur le corpus.

Le tableau des adpositions construit par notre script contient trop de données (au total 7553 lignes) pour que nous puissions vérifier manuellement si chaque adposition est régie. Nous nous appuyons donc sur la fréquence relative pour décider. Plus la fréquence d'apparition relative d'une adposition en compagnie d'un verbe est élevée, plus il est probable que cette

adposition soit régie par ce verbe. Il s'agit alors de déterminer un seuil de fréquence qui minimise le nombre d'erreurs (soit les adpositions qui ne sont pas régies) et maximise la capture des adpositions régies. Nous présenterons ci-dessous comment nous avons procédé pour trouver ce seuil.

Face à beaucoup de données, il faut prendre un échantillon représentatif pour pouvoir l'analyser manuellement. Avant de créer l'échantillon, il faut préciser que nous ne voulons pas perdre beaucoup d'adpositions régies. Si nous prenons un seuil très élevé, les adpositions régies avec une fréquence assez basse vont être éliminées. Pour cela, nous ne pouvons pas choisir un seuil très haut. C'est la raison pour laquelle nous avons décidé de créer un échantillon seulement pour les données avec une fréquence de moins de 0.5, au lieu de tout le tableau. La création de cet échantillon est faite à l'aide de scripts Python : nous choisissons au hasard le premier élément parmi les dix premières lignes des données déjà ordonnées par fréquence, et puis nous prenons les éléments à une distance de dix lignes. De cette façon, nous pouvons avoir un échantillon qui va être distribué un peu de façon uniforme à l'intérieur du tableau d'adpositions, À la fin, nous obtenons un échantillon ordonné sous forme de tableau, comme illustré au Tableau V.

Verbe	***Frame Element***	**Type**	**Adposition**	**Fréquence relative**
捆 'attacher'	Instrument	préposition	用 'avec'	0.458
刷 'brosser'	Instrument	—	—	0.429
辯論 'argumenter'	Intl_2	préposition	與 'avec'	0.411
苦惱 's'inquiéter'	Reason	préposition	為 'pour'	0.394
抛 'lancer'	Source	préposition	從 'à partir de'	0.375
相處 's'entendre'	Co-actor_2	préposition	與 'avec'	0.370
…	…	…	…	…

Tableau V. Extrait de l'échantillon aléatoire

Avec cet échantillon, nous commençons à faire le travail manuel. Nous vérifions chaque ligne et jugeons si cette adposition est vraiment régie, en ajoutant une colonne nommée «Annotation de régime» à côté de la colonne de fréquence. Si c'est positif, nous remplissons le tableau avec 1. À l'inverse, nous entrons 0. Le Tableau VI est un extrait de l'échantillon annoté.

Ensuite, nous produisons quelques graphes qui nous aident à analyser le tableau mentionné et puis à décider où est le seuil :

1) Un graphe de dispersion des «0» et des «1», qui nous montre la dispersion des positifs et des négatifs à mesure que la fréquence s'élève. Selon notre hypothèse, la dispersion des négatifs devrait être très dense à basses fréquences, et de moins en moins dense à hautes fréquences, tandis que celle de positifs devrait présenter une tendance inverse. Comme nous montre la Figure 53, la dispersion des positifs diffère de ce que nous attendions. Nous pensons que c'est la loi de Zipf qui est à l'oeuvre. La Figure 54 nous montre qu'il y a beaucoup plus de données de basses fréquences que de hautes fréquences. La concentration des «0» et des «1» dans les faibles fréquences que nous voyons à la Figure 53 est due au fait que la majorité des données de ce tableau sont concentrées dans les faibles fréquences. Par conséquent, le graphe de dispersion n'est pas le bon outil ici.

Loi de Zipf

> La loi de Zipf est une loi empirique qui concerne la fréquence des mots dans un texte. Elle déclare que, dans un corpus d'énoncés en langage naturel, la fréquence d'apparition d'un mot est inversement proportionnelle à son classement dans le tableau des fréquences. (Zipf, 1949)

Verbe	***Frame Element***	**Type**	**Adposition**	**Fréquence relative**	**Annotation de régime**
捆 ‘attacher’	Instrument	préposition	用 ‘avec’	0.458	1
刷 ‘brosser’	Instrument	—	—	0.429	1
辯論 ‘argumenter’	Intl_2	préposition	與 ‘avec’	0.411	1
苦惱 ‘s’inquiéter’	Reason	préposition	為 ‘pour’	0.394	1
抛 ‘lancer’	Source	préposition	從 ‘à partir de’	0.375	1
相處 ‘s’entendre’	Co-actor_2	préposition	與 ‘avec’	0.370	1
滑 ‘glisser’	Ground_Endpoint	préposition	至 ‘jusqu’à’	0.350	1
綁 ‘lier’	Instrument	préposition	以 ‘avec’	0.333	1
透露 ‘divulguer’	Medium	postposition	中 ‘au milieu de’	0.333	0
分別 ‘se séparer’	Phenomenon_Source	postposition	上 ‘sur’	0.333	0
…	…	…	…	…	…

Tableau VI. Extrait de l’échantillon annoté

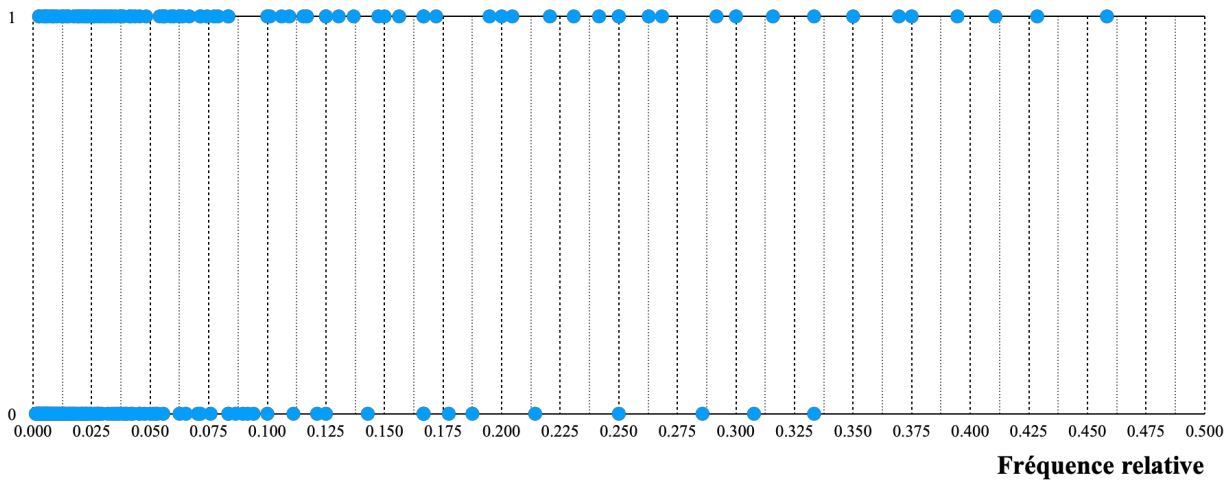


Figure 53. Graphe de dispersion des «0» et des «1»

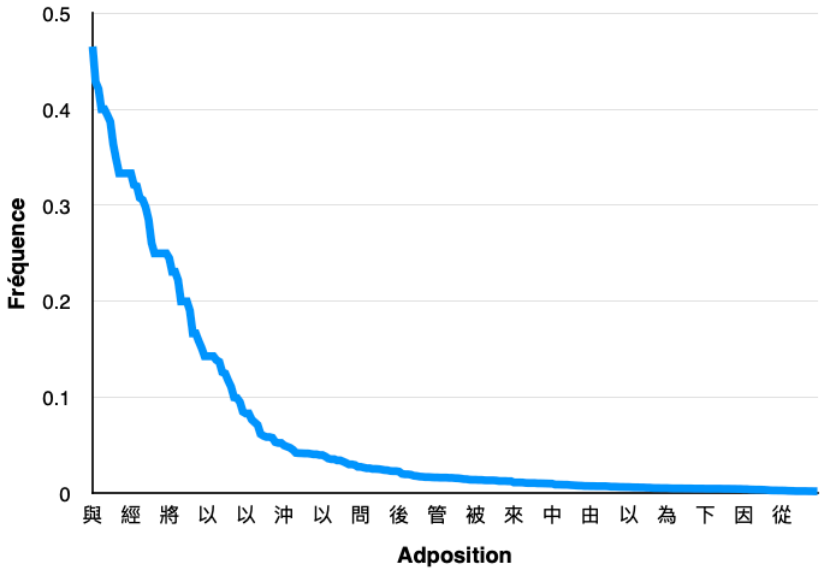


Figure 54. Graphe de la tendance des fréquences relatives

2) Un graphe du ratio de «1», qui nous illustre la tendance du ratio de positifs avec l'accroissement de la fréquence. Il faut mentionner que nous avons divisé la fréquence en 20 intervalles égaux, et que nous avons pris la valeur du milieu en tant que représentant de chaque intervalle. Les données préparées pour faire le graphe du ratio de «1» sont illustrées au Tableau VII. À la Figure 55, nous pouvons voir que le ratio de «1» présente une tendance générale ascendante, même s'il y a des points divergents. Dans ce graphe, nous ajoutons une courbe de tendance de prévision logarithme, en pointillé. De plus, l'équation de cette courbe de tendance est ajoutée en haut à droite, qui nous permet de calculer un seuil approximatif une fois que nous avons fixé le ratio de positifs. Cela nous donne une idée que le seuil ne serait pas une valeur fixe. C'est-à-dire que le choix de seuil dépend de l'objectif de l'utilisateur de ce corpus. Par exemple, si on veut avoir plus de positifs que de négatifs, on peut déterminer le ratio de «1» égale à 0.5, et puis utiliser l'équation pour obtenir le seuil correspondant; si on veut avoir peu de bruit, soit une précision élevée jusqu'à 90%, on peut déterminer le ratio de positif égale à 0.9 pour obtenir le seuil que l'on désire. En un mot, l'objectif de l'utilisation détermine en grande partie le seuil.

Intervalle de fréquence	**Représentant d'intervalle (l'axe X)**	**Nombre de données dans cet intervalle**	**Nombre de «1»**	**Ratio de «1» (l'axe Y)**
0-0.025	0.0125	352	26	0.074
0.025-0.05	0.0375	48	18	0.375
0.05-0.075	0.0625	25	14	0.560
0.075-0.1	0.0875	12	4	0.333
0.1-0.125	0.1125	11	7	0.636
0.125-0.15	0.1375	9	5	0.556
0.15-0.175	0.1625	10	5	0.500
0.175-0.2	0.1875	3	1	0.333
0.2-0.225	0.2125	5	4	0.800
0.225-0.25	0.2375	2	2	1.000
0.25-0.275	0.2625	9	6	0.667
0.275-0.3	0.2875	2	1	0.500
0.3-0.325	0.3125	3	2	0.667
0.325-0.35	0.3375	3	1	0.333
0.35-0.375	0.3625	2	2	1.000
0.375-0.4	0.3875	2	2	1.000
0.4-0.425	0.4125	1	1	1.000
0.425-0.45	0.4375	1	1	1.000
0.45-0.475	0.4625	1	1	1.000

Tableau VII. Données préparées pour le graphe du ratio de «1»

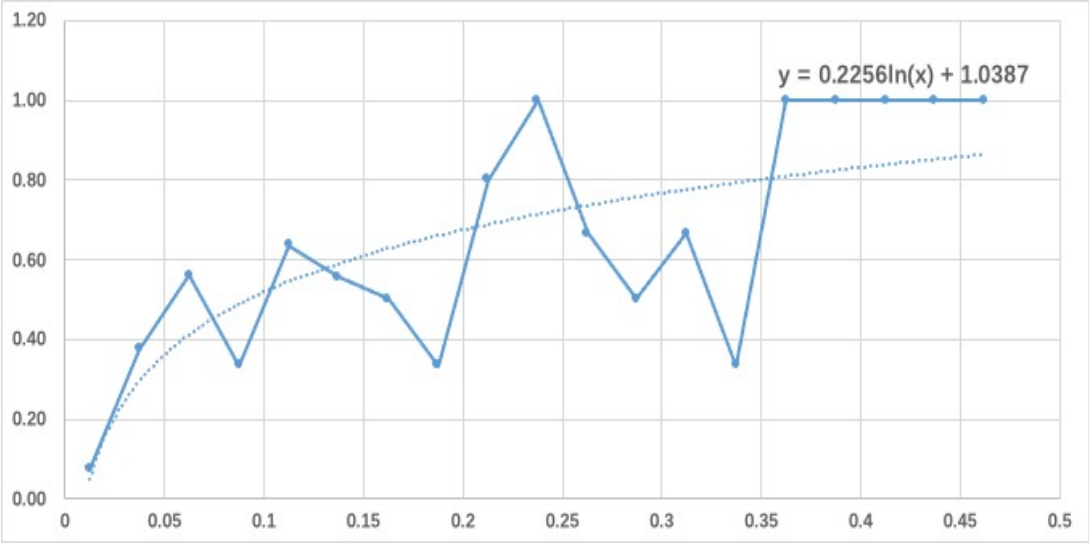


Figure 55. Graphe du ratio de «1»

3) Un graphe de comparaison qui contient quatre lignes représentant respectivement les tendances du ratio de «0» filtrés, du ratio de «1» filtrés, du ratio de «1» versus «0» et du ratio de données conservées, à mesure que le seuil augmente. Parmi les quatre groupes de données, seulement le dernier groupe est basé sur le tableau d'adpositions au lieu de l'échantillon. Les données préparées et le graphe sont illustrés au Tableau VIII et à la Figure 56. Ce graphe de comparaison aide à trouver des points intéressants ou des points qui correspondent au besoin de l'utilisateur d'une façon visuelle.

Seuil utilisé (l'axe X)	Nombre de «0» filtrés	Ratio de «0» filtrés	Nombre de «1» filtrés	Ratio de «1» filtrés	Ratio de «1» vs «0»	Nombre de données conservées	Ratio de données conservées
0.030	339	0.852	30	0.291	0.553	3858	0.511
0.035	344	0.864	35	0.340	0.557	3754	0.497
0.040	350	0.879	39	0.379	0.571	3656	0.484
0.045	353	0.887	42	0.408	0.575	3595	0.476
0.050	356	0.894	44	0.427	0.584	3544	0.469
0.055	362	0.910	45	0.437	0.617	3483	0.461
0.060	363	0.912	49	0.476	0.607	3424	0.453
0.065	364	0.915	53	0.515	0.595	3377	0.447
0.070	365	0.917	54	0.524	0.598	3355	0.444
0.075	367	0.922	58	0.563	0.592	3299	0.437
0.080	368	0.925	60	0.583	0.589	3271	0.433
0.085	371	0.932	62	0.602	0.603	3217	0.426
0.090	373	0.937	62	0.602	0.621	3202	0.424
0.095	375	0.942	62	0.602	0.641	3193	0.423
0.100	375	0.942	62	0.602	0.641	3176	0.420
0.125	379	0.952	69	0.670	0.642	3068	0.406

Tableau VIII. Données préparées pour le graphe de comparaison

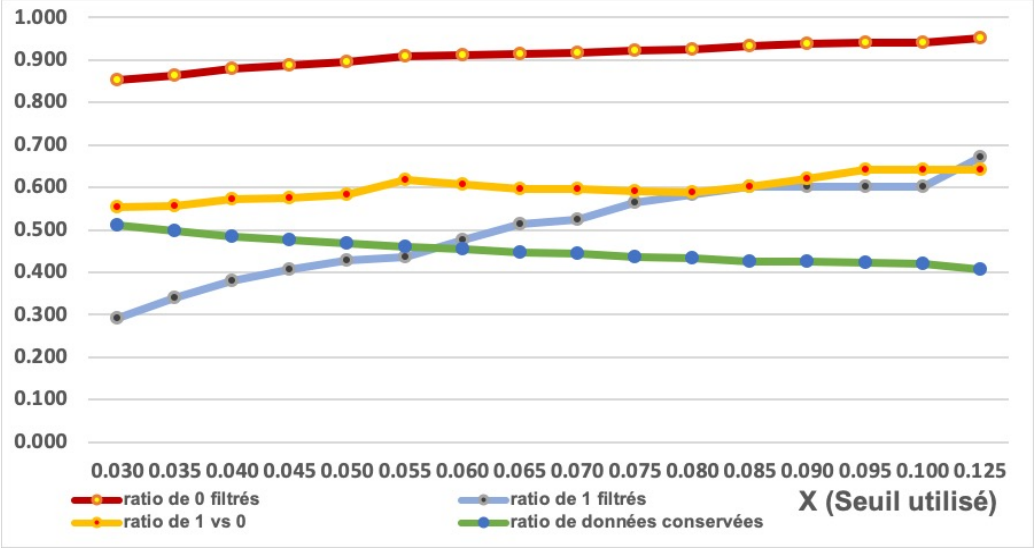


Figure 56. Graphe de comparaison

4.2 Dictionnaire final

Selon ce que nous avons présenté plus haut, la valeur du seuil peut varier en fonction de l'objectif de l'utilisateur. Par conséquent, les entrées dans le dictionnaire final ne sont pas figées, puisque le contenu de ce dictionnaire est ce qui reste après le filtrage selon un certain seuil choisi. Dans l'intention de faciliter la compilation du dictionnaire par l'utilisateur, nous avons créé une commande illustrée à la Figure 57, et nous offrons en même temps les fichiers nécessaires au fonctionnement de cette commande[21]. Cette commande génère un dictionnaire final sous forme de tableau en paramétrant le seuil qui correspond au besoin de l'utilisateur. Dans les paragraphes suivants, nous allons présenter la commande, les fichiers relatifs et le dictionnaire final généré.

```
python3 dict_rv_m-main/dic_v.py -c dict_rv_m-main/phrases/ -pr dict_rv_m-main/prep_s_tw_hk.txt -po dict_rv_m-main/post_s_tw_hk.txt -t 0.055
```

Figure 57. Commande pour créer un dictionnaire final (seuil égal à 0.055)

Cette commande appelle le script Python *dic_v.py*, qui réalise l'extraction de l'information utile de la ressource choisie (cf. §3.3) et la génération d'un dictionnaire final. En plus de cela, nous avons besoin également de quatre paramètres : *-c*, le fichier du corpus utilisé (cf. §3.1), *-pr* le fichier de la liste de prépositions (cf. §3.2), *-po*, le fichier de la liste de postpositions (cf. §3.2) et *-t*, le seuil déterminé (cf. §4.1). Il faut mentionner que les fichiers utilisés dans la commande sont tous offerts par nous, mais la valeur du seuil est laissée au choix de l'utilisateur.

Le dictionnaire final correspondant au seuil déterminé sera généré après l'exécution de la commande. Il contient seulement les éléments dont la fréquence d'apparition dépasse le seuil paramétré. Le Tableau IX nous illustre un extrait du dictionnaire final généré avec le seuil égal à 0.055. Nous prenons le verbe 說 shuō 'parler/dire' comme exemple pour expliquer l'organisation de ce dictionnaire :

1) Dans ce dictionnaire, le verbe 說 a cinq *frame elements*;

[21] https://github.com/a964913546/dict_rv_m

2) Parmi les cinq *frame elements*, il y en a trois qui s'expriment sans adposition : «Message_Description», «Message», et «Speaker»;

3) Le *frame element* «Addressee» peut s'exprimer sans adposition ou avec une de trois prépositions régies par le verbe;

4) Le *frame element* «Topic» peut s'exprimer sans adposition ou avec une préposition régie par le verbe;

5) Généralement, pour chaque *frame element* d'un verbe, la somme de la fréquence d'apparition de tous les cas doit être 1. À part les *frame elements* «Message» et «Topic», la somme de la fréquence des trois autres *frame elements* n'est pas 1. Cela signifie qu'il y a des cas qui ont été filtrés par le seuil.

Verbe	***Frame Element***	**Adposition**	**Fréquence relative**
說 'dire'	Message_Description		0.976
說	Addressee		0.556
說	Addressee	跟 'avex'	0.194
說	Addressee	給 'à'	0.056
說	Addressee	對 'à'	0.139
說	Message		1
說	Speaker		0.974
說	Topic		0.5
說	Topic	關於 'à propos de'	0.5

Tableau IX. Un extrait du dictionnaire final (seuil égal à 0.055)

Chapitre 5 Discussion : limites et travaux futurs

Après avoir généré le dictionnaire final, nous menons à terme notre recherche sur la création d'un dictionnaire de régimes verbaux en mandarin. Dans le fond, cela signifie un nouveau commencement de réflexion, d'amélioration et d'avancement. Dans cette section, nous allons discuter des problèmes que nous avons rencontrés au cours de la recherche et discuter de pistes de travail futur.

Après la création de l'échantillon du tableau d'adpositions, nous l'avons vérifié manuellement et ajouté une colonne appelée «Annotation de régime» (cf. Tableau VI). Au début, nous avions l'intention de faire faire ce travail par plusieurs locuteurs natifs. Malheureusement, cette idée n'était pas faisable, parce que ce travail exige des notions de sémantique et le grand nombre d'éléments à vérifier prend beaucoup de temps. Par conséquent, le travail de jugement a été fait par nous-même. Cela signifie que nous ne pouvons pas prétendre au même type de fiabilité fournie par un accord inter-annotateur.

Pourtant, cette annotation a un impact direct sur la sélection du seuil et sur la création du dictionnaire final. Autrement dit, le travail de jugement détermine en grande partie la précision du dictionnaire généré, puisque nous considérons toutes les adpositions dont la fréquence d'apparition dépasse le seuil choisi comme des adpositions régies par le verbe correspondant. C'est-à-dire, il y a aussi des erreurs apportées par la colonne «Annotation de régime», à part les erreurs existantes causées par le filtrage (soit les négatifs qui ont une fréquence d'apparitions excède le seuil).

En résumé, le travail de la colonne «Annotation de régime» est un point qui appellera une meilleure solution dans les améliorations futures. Dans le but de créer un dictionnaire ayant une précision plus haute, la recherche exige plus de travail manuel. Il faut peser la précision et les coûts de temps et de travail manuel.

Dans notre recherche, nous utilisons la fréquence relative pour analyser le tableau d'adposition. Ce moyen n'est pas assez précise, puisqu'il est probable que d'autres facteurs puissent contribuer à la fréquence relative, par exemple la fréquence globale d'une adposition, ou certaines propriétés sémantiques du verbe ou du rôle thématique considéré. C'est donc mieux

d'utiliser des moyens plus subtils que la fréquence relative, par exemple l'information mutuelle spécifique (PMI), qui tient compte en même temps de la fréquence de l'adposition apparaissant avec le lemme et la fréquence absolue de l'adposition et du lemme indépendamment dans le corpus.

Information mutuelle spécifique (PMI)

> La formule pour calculer le PMI est :
>
> $$\mathrm{PMI}(x, y) = \log \frac{p(x,y)}{p(x)p(y)}$$
>
> Dans la formule, $p(x,y)$ représente la probabilité que x et y apparaissent ensemble dans le corpus, $p(x)$ et $p(y)$ sont respectivement la probabilité de trouver indépendamment x et y dans le corpus.

Essayer d'autres techniques et puis faire une comparaison entre les résultats permettraient de raffiner notre travail et de trouver le manque ou le problème de notre recherche.

Au cours de cette recherche, nous avons trouvé que les ressources en mandarin pour la GAT sont rares. Comme nous n'avons pas trouvé de dictionnaire de type VerbNet qu'on puisse intégrer à GenDR, nous avons commencé la recherche actuelle. Le dictionnaire créé sert à favoriser l'ajout du mandarin dans GenDR. Par conséquent, l'intégration à proprement parler de ce dictionnaire de régimes verbaux en mandarin dans GenDR pourrait être le travail suivant, comme ce que Galarreta-Piquette (2018) a fait pour l'anglais.

L'usage d'un dictionnaire en TAL vise à assurer la précision. Dans ce cas, un dictionnaire doit avoir une très haute précision. Donc, l'amélioration de la précision du dictionnaire sera une tâche permanente. Ainsi, améliorer notre dictionnaire pour qu'il puisse être appliqué dans d'autres applications est une autre direction de travail futur.

La ressource lexicale sur laquelle notre recherche se base est un projet actif et continuellement mis à jour. Ainsi, de nouveaux verbes seront ajoutés de temps en temps dans le corpus. Le renouvellement de notre dictionnaire serait ainsi un travail possible à faire dans le futur.

Chapitre 6 Conclusion

Notre recherche vise à combler une lacune pour le mandarin dans le domaine de la GAT : créer un dictionnaire de régimes verbaux en mandarin qui contribue grandement à l'intégration du mandarin dans le projet GenDR. Nous avons réussi à atteindre les objectifs fixés au début de notre recherche :

1) Nous avons utilisé des scripts Python pour extraire les patrons de construction et les phrases d'exemple à partir de la base de données de *Mandarin VerbNet* et ensuite les représenter sous une forme qui peut être traitée ultérieurement;
2) Nous avons analysé les informations extraites et avons détecté des informations à propos des adpositions avec l'aide des scripts Python. Les informations détectées sont collectées sous forme de tableau. Après le travail manuel de jugement sur les adpositions prélevées, nous avons créé un dictionnaire adaptable selon l'objectif de son utilisateur.

Dans le premier chapitre, nous avons introduit la GAT et le projet GenDR, un réalisateur profond générique multilingue, et présenté la problématique qui nous intéresse. Le but final de créer un module GenDR spécifique pour le mandarin nous mène à la création d'un dictionnaire spécifique en mandarin. Nous nous sommes concentré sur les verbes, qui ont plus de variété dans leurs régimes que d'autres parties du discours. L'imprévisibilité du régime des verbes et le rôle central que les verbes jouent dans un énoncé nous conduisent à créer un dictionnaire de régimes verbaux.

Dans le deuxième chapitre, nous avons fait une mise en contexte des concepts liés à notre recherche. La TST est le cadre théorique linguistique sur laquelle le projet GenDR se base. Notre recherche a le but de servir GenDR en fournissant de l'information sur le régime des verbes en mandarin.

Dans le troisième chapitre, nous avons présenté en détail la méthodologie de notre recherche. À l'aide de scripts Python, nous avons extrait les adpositions de la ressource lexicale *Mandarin VerbNet*, en utilisant les listes d'adpositions préparées. Ces informations extraites mises en tableau attendent le traitement final pour devenir le dictionnaire.

Dans le quatrième chapitre, nous prenons un échantillon pour évaluer les données extraites. Il faut un travail manuel pour vérifier si chaque adposition extraite est vraiment régie par le verbe correspondant. Ensuite, nous avons recours à des graphes pour fixer un seuil qui détermine le contenu du dictionnaire généré. À la fin, il est clair que le seuil est choisi selon l'objectif de l'utilisateur du dictionnaire, au lieu d'être une valeur figée.

Dans le cinquième chapitre, nous réfléchissons sur la précision du dictionnaire final et sur la recherche possible à faire dans le futur. L'amélioration de la précision exige plus de coûts de temps et de travail manuel. Le travail futur est lié principalement à l'application dans les systèmes de GAT.

Bibliographie

Ayan, N. F., & Dorr, B. J. (2002). Generating A Parsing Lexicon from an LCS-Based Lexicon. *Proceedings of the LREC-2002 Workshop on Linguistic Knowledge Acquisition and Representation*, 4352.

Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998). The Berkeley FrameNet Project. *Proceedings of the 17th International Conference on Computational Linguistics - Volume 1*, 86-90. https://doi.org/10.3115/980451.980860

Bateman, J., & Zock, M. (2003). *Natural Language Generation*. In R. Mitkov (Ed.), *Oxford handbook of computational linguistics* (pp. 284–304). London: Oxford University Press, Chap. 15

Belz, A., & Kow, E. (2009). System Building Cost vs. Output Quality in Data-to-text Generation. *Proceedings of the 12th European Workshop on Natural Language Generation*, 16-24. http://dl.acm.org/citation.cfm?id=1610195.1610198

Bohnet, B., Langjahr, A., & Wanner, L. (2000). A Development Environment for an MTT-based Sentence Generator. *Proceedings of the First International Conference on Natural Language Generation - Volume 14*, 260-263. https://doi.org/10.3115/1118253.1118292

Bohnet, B., & Wanner, L. (2010). Open Soucre Graph Transducer Interpreter and Grammar Development Environment. *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. http://www.lrec-conf.org/proceedings/lrec2010/pdf/585_Paper.pdf

Bollmann, M. (2011). Adapting SimpleNLG to German. *Proceedings of the 13th European Workshop on Natural Language Generation*, 133-138. https://www.aclweb.org/anthology/W11-2817

Busso, L., & Lenci, A. (2016). Italian VerbNet : A Construction-based Approach to Italian Verb Classification. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2633-2642. https://www.aclweb.org/anthology/L16-1419

Carenini, G., Mittal, V. O., & Moore, J. D. (1994). Generating patient-specific interactive natural language explanations. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, 5-9.

CLEAR. (2005). *VerbNet Annotation Guidelines*.

Copestake, A. (1992). The ACQUILEX LKB: Representation Issues in Semi-Automatic Acquisition of Large Lexicons. *Proceedings of the Third Conference on Applied Natural Language Processing*, 88-95. https://doi.org/10.3115/974499.974515

Danlos, L. (1983). Présentation d'un modèle de génération automatique. *Revue québécoise de linguistique*, *13*(1), 203-228. https://doi.org/10.7202/602510ar

Danlos, L., Nakamura, T., & Pradet, Q. (2014, juillet). Vers la création d'un VerbeNet du français. *Atelier FondamenTAL, TALN 2014*.

Danlos, L., Nakamura, T., & Pradet, Q. (2015). Traduction de VerbNet vers le français. *Congrès ACFAS*, 1. https://hal.inria.fr/hal-01179175

Danlos, L., Pradet, Q., barque, L., Takuya, N., & Constant, M. (2016). Un Verbenet du français. *Traitement Automatique des Langue*, *57*(1), 33-58.

Daoust, N. (2014). *JSreal : Un réalisateur de texte pour la programmation web* [Mémoire de maîtrise, Université de Montréal].

Daoust, N., & Lapalme, G. (2015). JSREAL : A Text Realizer for Web Programming. In N. Gala, R. Rapp, & G. Bel-Enguix (Éds.), *Language Production, Cognition, and the Lexicon* (p. 361-376). Springer International Publishing. https://doi.org/10.1007/978-3-319-08043-7_21

de Marneffe, M.-C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., & Manning, C. D. (2014). Universal Stanford dependencies : A cross-linguistic typology. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 4585-4592. http://www.lrec-conf.org/proceedings/lrec2014/pdf/1062_Paper.pdf

de Marneffe, M.-C., MacCartney, B., & Manning, C. (2006). Generating Typed Dependency Parses from Phrase Structure Parses. *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, *6*. http://www.lrec-conf.org/proceedings/lrec2006/pdf/440_pdf.pdf

de Marneffe, M.-C., & Manning, C. D. (2008). The Stanford Typed Dependencies Representation. *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, 1-8. https://www.aclweb.org/anthology/W08-1301

de Oliveira, R., & Sripada, S. (2014). Adapting SimpleNLG for Brazilian Portuguese realisation. *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, 93-94. https://doi.org/10.3115/v1/W14-4412

Dorr, B. J. (2001). LCS Verb Database. *Online Software Database of Lexical Conceptual Structures and Documentation*, University of Maryland.

Dubinskaite, I. (2017). *Développement de ressources lituaniennes pour un générateur automatique de texte multilingue* [Mémoire de maîtrise, Université Grenoble Alpes]. https://doi.org/10.13140/RG.2.2.27254.91204

Dubois, J., & Dubois-Charlier, F. (1997). *Les verbes français*. Larousse.

Elman, J. L. (1990). Finding Structure in Time. *Cognitive Science*, *14*(2), 179-211. https://doi.org/10.1207/s15516709cog1402_1

Fellbaum, C. (1998). *WordNet : An Electronic Lexical Database*. The MIT Press. https://doi.org/10.7551/mitpress/7287.001.0001

Fillmore, C. J. (1968). The Case for Case. In E. Bach & R. T. Harms (Éds.), *Universals in Linguistic Theory* (p. 1-88). Holt, Rinehart and Winston.

Fillmore, C. J., Johnson, C. R., & Petruck, M. R. L. (2003). Background to Framenet. *International Journal of Lexicography*, *16*(3), 235-250. https://doi.org/10.1093/ijl/16.3.235

Galarreta-Piquette, D. (2018). *Intégration de VerbNet dans un réalisateur profond* [Mémoire de maîtrise]. Université de Montréal.

Gatt, A., & Krahmer, E. (2018). Survey of the State of the Art in Natural Language Generation : Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, *61*, 65-170. https://doi.org/10.1613/jair.5477

Goldberg, E., Driedger, N., & Kittredge, R. I. (1994). Using natural-language processing to produce weather forecasts. *IEEE Expert*, *9*(2), 45-53. https://doi.org/10.1109/64.294135

Grishman, R., Macleod, C., & Meyers, A. (1994). Comlex Syntax : Building a Computational Lexicon. *Proceedings of the 15th Conference on Computational Linguistics - Volume 1*, 268-272. https://doi.org/10.3115/991886.991931

Gross, M. (1975). *Méthodes en syntaxe : Régime des constructions complétives*. Hermann.

Herbst, T., Heath, D., Roe, I. F., Götz, D., & Klotz, M. (2004). *A Valency Dictionary of English, A Corpus- Based Analysis of the Complementation Patterns of English Verbs, Nouns and Adjectives*. De Gruyter Mouton. https://www.degruyter.com/view/title/14364

Jackendoff, R. S. (1972). *Semantic Interpretation in Generative Grammar*. MIT Press.

Kahane, S., & Polguère, A. (2001). Formal foundation of lexical functions. *Proceedings of the Workshop on Collocations at ACL 2001*, 8.

Korhonen, A., Krymolowski, Y., & Briscoe, T. (2006). A Large Subcategorization Lexicon for Natural Language Processing Applications. *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. http://www.lrec-conf.org/proceedings/lrec2006/pdf/558_pdf.pdf

Lambrey, F. (2017). *Implémentation des collocations pour la réalisation de texte multilingue* [Mémoire de maîtrise]. Université de Montréal.

Lambrey, F., & Lareau, F. (2015). Le traitement des collocations en génération de texte multilingue. *Actes de la 22e conférence sur le Traitement Automatique des Langues Naturelles*, 263-269. https://www.aclweb.org/anthology/2015.jeptalnrecital-court.39

Langkilde, I. (2000). Forest-based Statistical Sentence Generation. *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, 170-177. http://dl.acm.org/citation.cfm?id=974305.974328

Lareau, F. (2002). *La synthèse automatique de paraphrases comme outil de vérification des dictionnaires et grammaires de type Sens-Texte* [Mémoire de maîtrise]. Université de Montréal.

Lareau, F., Lambrey, F., Dubinskaite, I., Galarreta-Piquette, D., & Nejat, M. (2018). GenDR : A Generic Deep Realizer with Complex Lexicalization. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*. LREC 2018, Miyazaki, Japan.

Lareau, F., & Wanner, L. (2007). Towards a generic multilingual dependency grammar for text generation. In T. H. King & E. M. Bender (Éds.), *Proceedings of the Grammar Engineering Across Frameworks (GEAF07) Workshop* (p. 203-223). CSLI Publications.

Lavoie, B., & Rainbow, O. (1997). A Fast and Portable Realizer for Text Generation Systems. *Fifth Conference on Applied Natural Language Processing*, 265-268. https://doi.org/10.3115/974557.974596

Leclère, C. (1990). Organisation du lexique-grammaire des verbes français. *Langue Française*, *87*, 112-122.

Levin, B. (1993). *English verb classes and alternations : A preliminary investigation*. The University of Chicago Press.

Liu, M.-C., & Chiang, T.-Y. (2008). The construction of Mandarin VerbNet : A frame-based study of statement verbs. *Language and Linguistics*, *9*(2), 239-270.

Mazzei, A., Battaglino, C., & Bosco, C. (2016). SimpleNLG-IT : Adapting SimpleNLG to Italian. *Proceedings of the 9th International Natural Language Generation conference*, 184-192. https://doi.org/10.18653/v1/W16-6630

Mel′čuk, I. (2007). Semantic Transition Rules (of the Semantic Module of the Meaning-Text Linguistic Model). *Proceedings of MTT 2007*.

Mel'čuk, I. (1988). *Dependency syntax : Theory and practice*. State University of New York Press.

Mel'čuk, I. (1995). The future of the lexicon in linguistic description and the explanatory combinatorial dictionary. In I.-H. LEE, *Linguistics in the morning calm* (Vol. 3). Hanshin.

Mel'čuk, I. (2001). *Communicative Organization in Natural Language : The semantic-communicative structure of sentences*. John Benjamins. https://doi.org/10.1075/slcs.57

Mel'čuk, I. (2012). *Semantics : From meaning to text* (Vol. 1). John Benjamins Publishing Company. https://benjamins.com/catalog/slcs.129

Mel'čuk, I. (2015). *Un modèle linguistique fonctionnel : Le modèle Sens-Texte*. INaLCO.

Mel'čuk, I. (1997). *Vers une linguistique Sens-Texte*. Leçon inaugurale (faite le Vendredi 10 janvier 1997), Collège de France, Chaire internationale, 43 pages.

Mel'čuk, I., & Milićević, J. (2014). *Introduction à la linguistique* (Vol. 1-3). Hermann.

Mellish, C., & Dale, R. (1998). Evaluation in the context of natural language generation. *Computer Speech & Language*, *12*(4), 349-373. https://doi.org/10.1006/csla.1998.0106

Milićević, J. (2006). *A Short Guide to the Meaning-Text Linguistic Theory*.

Milićević, J. (2007). *La paraphrase. Modélisation de la paraphrase langagière*. Peter Lang. https://www.peterlang.com/view/title/10282

Mille, S., Carlini, R., Burga, A., & Wanner, L. (2017). FORGe at SemEval-2017 Task 9 : Deep sentence generation based on a sequence of graph transducers. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 920-923. https://doi.org/10.18653/v1/S17-2158

Mille, S., & Wanner, L. (2017). A demo of FORGe : The Pompeu Fabra Open Rule-based Generator. *Proceedings of the 10th International Conference on Natural Language Generation*, 245-246. https://doi.org/10.18653/v1/W17-3539

Molins, P., & Lapalme, G. (2015). JSrealB : A Bilingual Text Realizer for Web Programming. *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, 109-111. https://doi.org/10.18653/v1/W15-4719

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., & Zeman, D. (2016). Universal Dependencies v1 : A Multilingual Treebank Collection. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 1659-1666. https://www.aclweb.org/anthology/L16-1262

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU : A Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 311-318. https://doi.org/10.3115/1073083.1073135

Paroubek, P., Schabes, Y., & Joshi, A. K. (1992). XTAG: A Graphical Workbench for Developing Tree-Adjoining Grammars. *Proceedings of the Third Conference on Applied Natural Language Processing*, 223-230. https://doi.org/10.3115/974499.974538

Petrov, S., Das, D., & McDonald, R. (2012). A Universal Part-of-Speech Tagset. *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, 2089-2096. http://www.lrec-conf.org/proceedings/lrec2012/pdf/274_Paper.pdf

Polguère, A. (1998). La théorie Sens-Texte. *Dialangue*, *8-9*, 9-30.

Pradet, Q., Danlos, L., & de Chalendar, G. (2014, mai). Adapting VerbNet to French using existing resources. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*.

Rambow, O., & Korelsky, T. (1992). Applied Text Generation. *Third Conference on Applied Natural Language Processing*, 40-47. https://doi.org/10.3115/974499.974508

Ramos-Soto, A., Janeiro-Gallardo, J., & Bugarín Diz, A. (2017). Adapting SimpleNLG to Spanish. *Proceedings of the 10th International Conference on Natural Language Generation*, 144-148. https://doi.org/10.18653/v1/W17-3521

Reiter, E., & Dale, R. (2000, janvier). *Building Natural Language Generation Systems*. Cambridge Core. https://doi.org/10.1017/CBO9780511519857

Schuler, K. K. (2005). *VerbNet : A broad-coverage, comprehensive verb lexicon* [PhD Thesis, University of Pennsylvania]. https://repository.upenn.edu/dissertations/AAI3179808

Tarasov, D. (2015). Natural Language Generation, Paraphrasing and Summarization of User Reviews with Recurrent Neural Networks. *Natural Language Generation*, *1*(14), 595-602.

Thompson, H. (1977). Strategy and tactics : A model for language production. *13th Regional Meeting of the Chicago Linguistics Society*.

Vaudry, P.-L., & Lapalme, G. (2013). Adapting SimpleNLG for Bilingual English-French Realisation. *Proceedings of the 14th European Workshop on Natural Language Generation*, 183-187. https://www.aclweb.org/anthology/W13-2125

Wanner, L. (1996). *Lexical Functions in Lexicography and Natural Language Processing*. John Benjamins Publishing Company. https://benjamins.com/catalog/slcs.31

Wanner, L. (1992). Lexical Choice and the Organization of Lexical Resources in Text Generation. *Proceedings of the 10th European Conference on Artificial Intelligence*, 495-499. http://dl.acm.org/citation.cfm?id=145448.146820

Wanner, L., & Bateman, J. A. (1990). A collocational based approach to salience-sensitive lexical selection. *Proceedings of the Fifth International Workshop on Natural Language Generation*. https://www.aclweb.org/anthology/W90-0105

Wanner, L., Bohnet, B., Bouayad-Agha, N., Lareau, F., & Nicklaß, D. (2010). Marquis : Generation of User-Tailored Multilingual Air Quality Bulletins. *Applied Artificial Intelligence*, *24*(10), 914-952. https://doi.org/10.1080/08839514.2010.529258

XTAG Research Group. (2001). *A Lexicalized Tree Adjoining Grammar for English* (cahier de recherche IRCS-01-03). University of Pennsylvania.

Zeman, D. (2008, mai). Reusable Tagset Conversion Using Tagset Drivers. *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. http://www.lrec-conf.org/proceedings/lrec2008/pdf/66_paper.pdf

Zipf, G. K. (1949). *Human Behavior And The Principle Of Least Effort*. Cambridge, Massachusetts: Addison-Wesley.