Université de Montréal

Lexicalisation souple en réalisation de texte

Par
Avril Gazeau

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

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

Août 2023

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

---

Ce mémoire intitulé

Lexicalisation souple en réalisation de texte

Présenté par

Avril Gazeau

A été évalué par un jury composé des personnes suivantes

Antoine Venant
Président-rapporteur

François Lareau
Directeur de recherche

Alain Polguère
Université de Lorraine, CNRS, ATILF
Membre du jury

Résumé

GenDR est un réalisateur de texte symbolique qui prend en entrée un graphe, une représentation sémantique, et génère les graphes sous forme d'arbres de dépendances syntaxiques lui correspondant. L'une des tâches de GenDR lui permettant d'effectuer cette transduction est la lexicalisation profonde. Il s'agit de choisir les bonnes unités lexicales exprimant les sémantèmes de la représentation sémantique d'entrée. Pour ce faire, GenDR a besoin d'un dictionnaire sémantique établissant la correspondance entre les sémantèmes et les unités lexicales correspondantes dans une langue donnée.

L'objectif de cette étude est d'élaborer un module de lexicalisation souple construisant automatiquement un dictionnaire sémantique du français riche pour GenDR, son dictionnaire actuel étant très pauvre. Plus le dictionnaire de GenDR est riche, plus sa capacité à paraphraser s'élargit, ce qui lui permet de produire la base de textes variés et naturels correspondant à un même sens. Pour y parvenir, nous avons testé deux méthodes.

La première méthode consistait à réorganiser les données du *Réseau Lexical du Français* sous la forme d'un dictionnaire sémantique, en faisant de chacun de ses noeuds une entrée du dictionnaire et des noeuds y étant reliés par un type de lien lexical que nous appelons *fonctions lexicales paradigmatiques sémantiquement vides* ses lexicalisations.

La deuxième méthode consistait à tester la capacité d'un modèle de langue neuronal contextuel à générer des lexicalisations supplémentaires potentielles correspondant aux plus proches voisins du vecteur calculé pour chaque entrée du dictionnaire afin de l'enrichir.

Le dictionnaire construit à partir du Réseau lexical du français est compatible avec GenDR et sa couverture a été considérablement élargie. L'utilité des lexicalisations supplémentaires générées par le modèle neuronal s'est avérée limitée, ce qui nous amène à conclure que le modèle testé n'est pas tout à fait apte à accomplir le genre de tâche que nous lui avons demandée.

Mots-clés: réalisation automatique de texte, interface sémantique-syntaxe, lexicalisation, plongements lexicaux

Abstract

GenDR is an automatic text realiser. Its input is a graph; a semantic representation, and its output is the corresponding syntactic dependencies tree graphs. One of GenDR's tasks to operate this transduction successfully is called deep lexicalization, i.e. choosing the right lexical units to express the input semantic representation's semantemes. To do so, GenDR needs access to a semantic dictionnary that maps the semantemes to the corresponding lexical units in a given language.

This study aims to develop a flexible lexicalization module to build a rich French semantic dictionary automatically for GenDR, its current one being very poor. The more data the semantic dictionary contains, the more paraphrases GenDR is able to produce, which enables it to generate the basis for natural and diverse texts associated to a same meaning. To achieve this, we have tested two different methods.

The first one involved the reorganization of the *French Lexical Network* in the shape of a semantic dictionary, by using each of the network's nodes as a dictionary entry and the nodes linked to it by a special lexical relationship we call *semantically empty paradigmatic lexical functions* as its lexicalizations.

The second method involved testing a contextual neural language model's ability to generate potential additional lexicalizations by calculating the vector of each of the dictionary entries and generating its closest neighbours in order to expand the semantic dictionary's coverage.

The dictionary we built from the data contained in the French Lexical Network is compatible with GenDR and its coverage has been significantly broadened. Use of the additional lexicalizations produced by the language model turned out to be limited, which brings us to the conclusion that the tested model isn't completely able to perform the task we've asked from it.

Keywords: automatic text realization, syntax-semantics interface, lexicalization, word embeddings

Table des matières

Liste des tableaux

Liste des figures

Liste des sigles et abréviations

FL	fonction lexicale
TST	théorie Sens-Texte
FLPSV	fonction lexicale paradigmatique sémantiquement vide
PAM	paramètre d'approximation maximale
LEC	Lexicologie Explicative et Combinatoire
RL-fr	Réseau Lexical du Français
DS	dictionnaire sémantique
MLM	*masked language modeling*
NSP	*next sentence prediction*
RSém	représentation sémantique
RSyntP	représentation syntaxique profonde
RSyntS	représentation syntaxique de surface
RMorphP	représentation morphologique profonde
RMorphS	représentation morphologique de surface
RPhonP	représentation phonologique profonde
RPhonS	représentation phonologique de surface

Remerciements

Je tiens d'abord à adresser mes remerciements les plus sincères à mon directeur de recherche, François Lareau, qui m'a accompagnée tout au long de ce mémoire avec patience et humour. Moi qui ne connaissais rien au traitement automatique des langues en commençant cette maîtrise, je ressors de cette expérience chargée de nouvelles connaissances grâce à la confiance que François m'a accordée.

Ensuite, je souhaite remercier mes cher·e·s collègues de l'OLST pour leur présence, leur écoute et leur motivation contagieuse : Naïma, Li, Fiona, Yutaka, Kaori, Youyang et Pauline. Les dîners et les sorties auront généré automatiquement de très beaux moments dans cette aventure.

Je remercie aussi mes parents, dont les encouragements et l'intérêt pour ma recherche n'ont jamais vacillé, bien que mon domaine ne leur soit pas familier.

Je tiens également à souligner l'apport financier du CRSH, qui m'a permis de me donner corps et âme dans ma recherche.

Finalement, je remercie toutes les personnes qui ont travaillé à développer GenDR avant moi, sans qui la présente recherche n'aurait pas pu voir le jour.

Chapitre 1

Introduction

1.1 Problématique

GenDR [1] (Lareau *et al.*, 2018) est un réalisateur de texte profond symbolique multilingue qui prend en entrée une représentation sémantique (RSém) abstraite et génère les représentations syntaxiques correspondantes dans une langue donnée. L'une des premières tâches de GenDR dans ce processus est de choisir les bonnes lexies pour exprimer le sens de la RSém donnée en entrée, une tâche qu'on appelle lexicalisation. Pour lexicaliser une RSém, GenDR a besoin de données linguistiques faisant correspondre les sens aux unités lexicales qui les expriment dans une langue donnée. Ces données se présentent dans GenDR sous forme de dictionnaire sémantique dont chaque entrée est un sémantème et chaque valeur est l'ensemble des lexies qui y correspondent. Plus il y a de lexicalisations possibles pour un sémantème, plus grand sera le nombre de paraphrases que le système sera en mesure de générer. Cette capacité à paraphraser est primordiale pour pouvoir générer des textes naturels, fluides et peu répétitifs.

Des dictionnaires sémantiques de base ont été compilés pour l'anglais (Galarreta-Piquette, 2018), le chinois (Zhao, 2018), le français (Lareau *et al.*, 2018), le lituanien (Dubinskaitė, 2017) et le persan (Lareau *et al.*, 2018), mais ils présentent tous deux problèmes :

1. Le nombre d'entrées qu'ils contiennent est très limité, et augmenter manuellement leur couverture serait très coûteux.
2. On y trouve des erreurs et des écarts d'uniformité puisqu'ils ont été compilés sans coordination par différentes personnes.

1. Le nom GenDR vient de *generic deep realizer*.

Notre recherche vise à résoudre ces problèmes de trois façons :

1. En construisant automatiquement un DS riche à partir de ressources existantes. L'automatisation du traitement des données linguistiques permet d'en traiter un plus grand volume et d'éviter les erreurs et les incohérences.
2. En exploitant les modèles de langue neuronaux pour augmenter la couverture du DS.
3. En ajoutant au module de lexicalisation chargé de produire le DS un paramètre précédemment absent permettant de régler la distance sémantique maximale entre les différentes lexicalisations. Cela permet une lexicalisation plus souple, tout en laissant l'utilisateur choisir le degré d'exactitude du texte qu'il souhaite générer par rapport à la représentation sémantique d'entrée.

Dans ce mémoire, nous travaillons sur le français, mais la méthode proposée est assez générique pour produire un DS dans n'importe quelle langue, du moment que les ressources nécessaires sont disponibles.

1.2 Organisation du mémoire

Ce mémoire est organisé comme suit :

1. Dans le présent chapitre, nous présentons le cadre théorique dans lequel nous avons mené notre recherche ainsi que GenDR, le réalisateur de texte profond multilingue qui en bénéficiera.
2. Dans le chapitre 2, nous présentons la ressource lexicographique à partir de laquelle nous avons construit le DS, notre méthode pour y arriver et l'évaluation du résultat final.
3. Dans le chapitre 3, nous présentons les plongements lexicaux et les méthodes que nous avons testées pour enrichir le DS en exploitant ceux-ci.
4. Dans le chapitre 4, nous présentons une évaluation exhaustive des résultats obtenus pour l'expérience présentée au chapitre 3.
5. Pour conclure, nous faisons un retour sur les objectifs de la recherche et suggérons des pistes de travaux futurs.

1.3 La théorie Sens-Texte

Cette recherche s'inscrit dans le cadre de la théorie Sens-Texte (TST) (Žolkovskij et Mel'čuk, 1967; Polguère, 1998b; Kahane, 2003; Milićević, 2006; Mel'čuk, 2012, 2016). Ce cadre théorique développé depuis les années soixante vise à décrire les langues naturelles et présente un formalisme se prêtant particulièrement bien aux applications computationnelles de la linguistique.

La TST modélise les langues naturelles dans le sens de la synthèse, c'est-à-dire qu'elle décrit comment les locuteurs d'une langue L parviennent à exprimer un sens S. En effet, selon Mel'čuk (2016), la production d'un énoncé est une action plus purement linguistique que sa compréhension, cette dernière étant fortement influencée par les connaissances du monde des locuteurs, les phénomènes pragmatiques, etc. La description de la langue du sens vers le texte implique en son centre les paraphrases, c'est-à-dire tous les textes correspondant à un même sens. En effet, en TST, le sens est défini comme le point commun partagé par un ensemble de paraphrases, et est le point de départ de la synthèse du texte.

Pour modéliser les langues naturelles, la TST fait appel à un modèle Sens-Texte (Mel'čuk, 1981). Un modèle Sens-Texte établit la correspondance entre une représentation formelle du sens et une représentation formelle du texte, respectivement la représentation sémantique (RSém) et la représentation phonologique (RPhon). Selon Mel'čuk (2016), un modèle Sens-Texte sert aussi à tester la validité de la TST en générant toutes les paraphrases possibles pour une représentation sémantique, d'où l'objectif d'améliorer la capacité à paraphraser de GenDR, lui-même basé sur la TST (§1.4). Un modèle possède également plusieurs niveaux de représentations intermédiaires permettant le passage du sens vers le texte. Les sept niveaux de représentation linguistique de la TST sont listés dans l'ordre du sens vers le texte ci-dessous :

1. La représentation sémantique (RSém)
2. La représentation syntaxique, divisée en deux sous-niveaux :
   (a) La représentation syntaxique profonde (RSyntP)
   (b) La représentation syntaxique de surface (RSyntS)
3. La représentation morphologique, divisée en deux sous-niveaux :
   (a) La représentation morphologique profonde (RMorphP)
   (b) La représentation morphologique de surface (RMorphS)
4. La représentation phonologique, divisée en deux sous-niveaux :

(a) La représentation phonologique profonde (RPhonP)

(b) La représentation phonologique de surface (RPhonS)

Pour passer d'un niveau à l'autre, un modèle Sens-Texte possède un module de règles établissant la correspondance entre chaque niveau de représentation et le suivant. En partant d'une RSém, un modèle Sens-Texte produit une RPhonS, soit un énoncé linguistique.

La figure 1.1 présente visuellement l'architecture d'un modèle Sens-Texte.

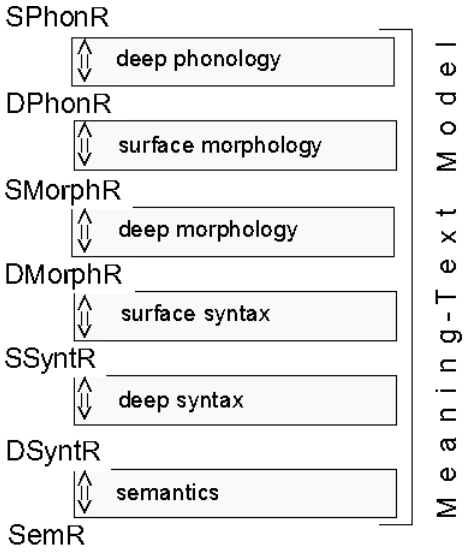


FIGURE 1.1 – Architecture d'un modèle Sens-Texte (Milićević, 2006)

Dans la section suivante, nous présentons GenDR, qui met en application les principes de la TST.

1.4 GenDR

GenDR est un réalisateur de texte automatique profond multilingue basé sur un transducteur de graphe appelé MATE (Bohnet *et al.*, 2000; Bohnet et Wanner, 2010). Il est l'héritier d'un système de diffusion de bulletins météo automatisé appelé MARQUIS (Wanner *et al.*, 2007). Ces trois systèmes sont basés sur la TST. L'architecture de GenDR est composée de dictionnaires, d'un éditeur de graphe et d'un éditeur de règles. GenDR n'opère pas dans tous les niveaux de représentation linguistique présents en TST, mais seulement aux niveaux sémantique, syntaxique profond et syntaxique de surface. Essentiellement, GenDR prend en entrée une RSém et retourne une RSyntS, en passant par une RSyntP. Les deux tâches principales permettant cette transition sont l'arborisation et la lexicalisation.

Les sections qui suivent décrivent l'architecture de GenDR ainsi que les tâches d'arborisation et de lexicalisation.

1.4.1 Les graphes

La représentation sémantique

La RSém est la structure d'entrée pour GenDR. Elle est formellement représentée par un réseau sémantique composé de nœuds et d'arcs. Les nœuds représentent les unités sémantiques présentes dans une langue, aussi appelées sémantèmes, et les arcs représentent les relations prédicat-argument qu'ils entretiennent entre eux. Il n'existe en TST que trois types de sémantèmes (Mel'čuk et Polguère, 2008; Mel'čuk, 2012, 2016) :

Les prédicats sémantiques dénotent des actions, des événements, des relations, etc. Pour prendre leur sens, ils doivent être associés à des arguments. Les arguments des prédicats sémantiques sont appelés leurs actants sémantiques. Le nombre d'actants qu'un prédicat peut accueillir est fixe. Par exemple, le prédicat 'dormir' ne peut accueillir qu'un seul actant (X dort), le prédicat 'manger' en possède deux (X mange Y), 'donner' en possède trois (X donne Y à Z), etc. C'est à partir des prédicats et des relations qu'ils entretiennent avec leurs actants que l'on construit les représentations sémantiques.

Les quasi-prédicats possèdent des propriétés à la fois des prédicats et des noms sémantiques. En effet, ils dénotent des entités plutôt que des faits, mais sont impliqués dans des actions ou des relations. Par exemple, le quasi-prédicat 'mère' possède la structure argumentale 'mère de X', car 'mère' implique par définition un participant sous-jacent dans la relation dénotée par le quasi-prédicat.

Les noms sémantiques dénotent des entités plutôt que des faits. Leur sens ne dépend pas d'un quelconque nombre d'arguments, il existe en soi. Un nom sémantique ne peut d'ailleurs pas accueillir d'arguments. Les noms sémantiques peuvent dénoter des objets ('caillou'), des lieux ('désert'), des êtres vivants ('arbre'), etc.

Dans la RSém, les arcs reliant les prédicats à leurs arguments sont numérotés selon leur position logique (Mel'čuk, 2004) et ne possèdent pas de sens.

Dans GenDR, le sens le plus saillant de la RSém est marqué pour être ensuite traduit comme la racine syntaxique de la RSyntP. La figure 1.2 présente une RSém qu'on pourrait donner en entrée à GenDR.

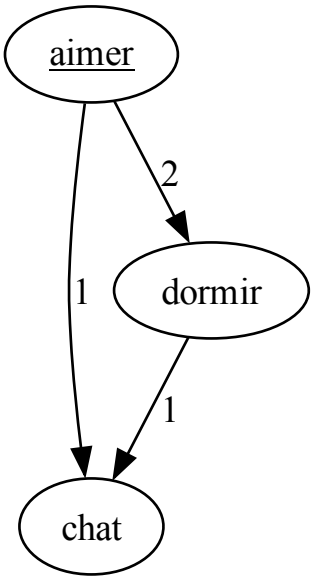


FIGURE 1.2 – Représentation sémantique

La RSyntP

La RSyntP suit la même logique que la RSém et est composée, elle aussi, de nœuds et d'arcs. Toutefois, elle n'est plus sous forme de réseau mais plutôt d'arbre de dépendances syntaxiques (Tesnière, 1959) dans lequel les nœuds correspondent à des lexies sémantiquement pleines et les branches aux relations syntaxiques qui les lient.

Les arcs encodent toujours les relations actancielles entre les prédicats et leurs arguments, mais se précisent à ce niveau de représentation. En effet, les relations prédicat-argument présentes au niveau sémantique se sous-divisent en deux types majeurs de relations syntaxiques (Mel'čuk, 2004) :

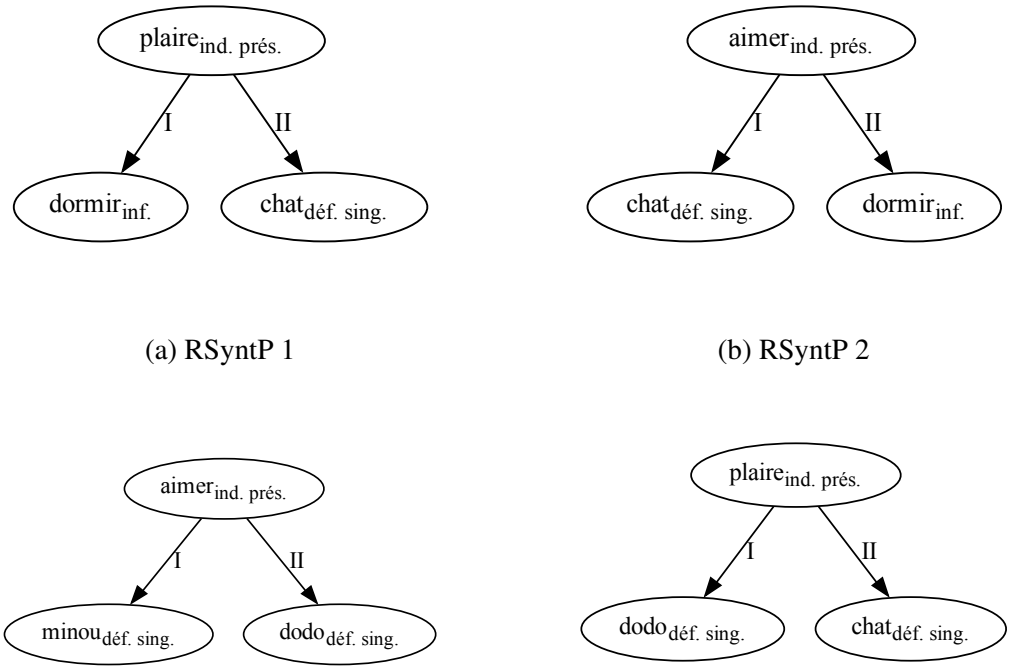
1. Les actants correspondent en syntaxe profonde aux dépendants syntaxiques d'une lexie **L** déterminés par son patron de régime et expriment les actants sémantiques du sémantème correspondant à **L**. Ils sont numérotés par des chiffres romains allant de I à VI.
2. Les modificateurs représentent tout type de relation de modification, et ne sont pas imposés par le patron de régime d'une lexie **L**.

Des grammèmes sont attachés aux nœuds de la RSyntP. Ils représentent des valeurs

flexionnelles sémantiquement pleines. Pour les noms, il s'agit par exemple de valeurs de définitude ou de nombre. Pour les verbes, il s'agit plutôt des valeurs de temps, de mode et d'aspect. Par exemple, le nœud CHAT DÉF, PL correspond à la séquence *les chats* en français, et le nœud DORMIR SUBJ, PRÉS correspond à *dorme*.

La racine syntaxique de la RSyntP correspond au sens le plus saillant de la RSém. En français, dans GenDR, cette racine doit être un verbe conjugué au mode indicatif.

La figure 1.3 présente des RSyntP correspondant à la RSém présentée en 1.2.



(a) RSyntP 1

(b) RSyntP 2

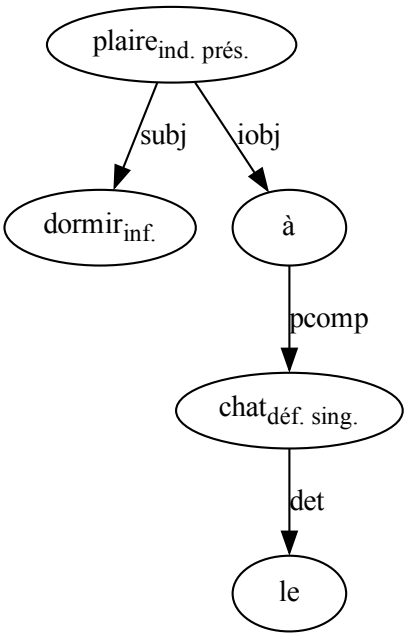
(c) RSyntP 3

(d) RSyntP 4

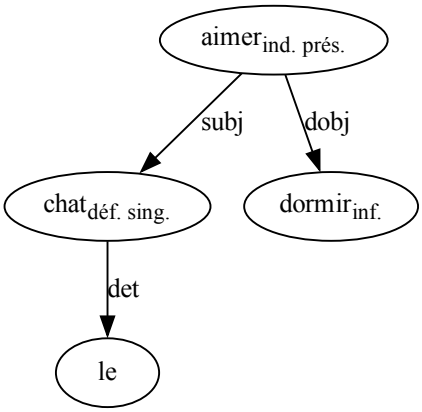
FIGURE 1.3 – RSyntP possibles pour la SSém 1.2

La RSyntS

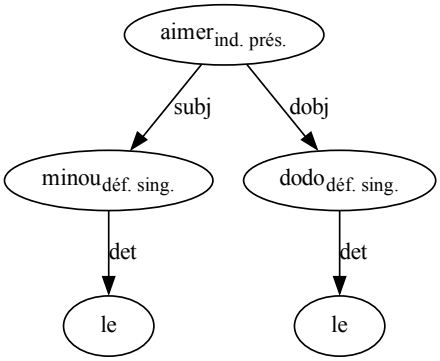
La RSyntS, quant à elle, est un arbre de dépendance syntaxique dont les branches sont étiquetées par des noms de fonctions syntaxiques plutôt que des chiffres, et dans lequel les unités lexicales fonctionnelles sont intégrées. Les nœuds sémantiquement pleins de la RSyntP et leurs grammèmes sont recopiés dans la RSyntS. La figure 1.4 présente les RSyntS correspondant aux RSyntP de la figure 1.3.



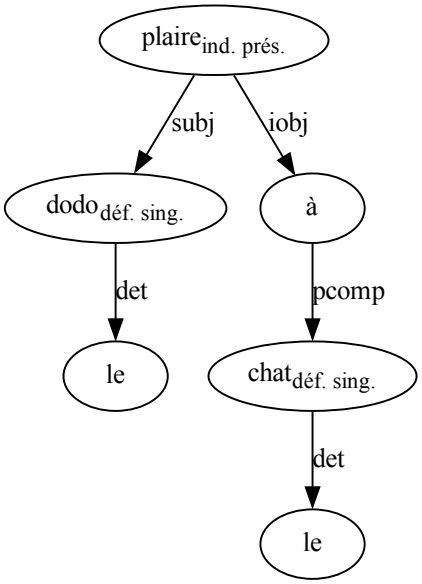
(a) RSyntS 1



(b) RSyntS 2



(c) RSyntS 3



(d) RSyntS 4

FIGURE 1.4 – RSyntS pouvant exprimer les RSyntP en 1.3

1.4.2 Les dictionnaires

Le dictionnaire sémantique (DS)

Le DS est au centre de cette recherche. En effet, c'est cet élément de l'architecture de GenDR que nous cherchons à enrichir. Les entrées du DS sont des sémantèmes auxquels est associé l'ensemble des lexicalisations profondes correspondantes dans la langue. Ce dictionnaire permet la mise en correspondance des nœuds de la RSém et ceux de la RSyntP.

À l'heure actuelle, le DS du français de GenDR contient environ 1425 sémantèmes correspondant à environ 1555 lexicalisations.

Le listing 1.1 présente la structure des données du DS.

Le dictionnaire lexical

Le dictionnaire lexical comporte de l'information morpho-syntaxique au sujet de chacune de ses entrées. Chaque lexicalisation associée à une entrée dans le DS doit avoir son entrée dans le dictionnaire lexical. Les informations associées à une entrée du dictionnaire lexical incluent :

— la partie du discours ;

— la diathèse ;

— le patron de régime ;

— les fonctions lexicales (FL) qui s'y appliquent.

Lors de la transduction de la RSém vers la RSyntP, GenDR visite les entrées du dictionnaire lexical correspondant aux lexicalisations des nœuds de la RSém et en retrouve les informations syntaxiques nécessaires pour générer des RSyntP adéquates.

Listing 1.1 – Exemple d'entrée dans le DS

```
"précis_Adj#1" {
    lex="clair_Adj#III"
    lex="distinct#II"
    lex="net_Adj#III.2"
    lex="précis_Adj#1"
    lex="précision"
    lex="précisément"
}
```

Listing 1.2 – Exemple d'entrée dans le dictionnaire lexical

```
exposé : noun {
    dpos = N
    spos = noun
    gp = {
        1 = I
        2 = II
        3 = III {dpos=N prep=sur}
    }
    lf = {name=Oper2 value=assister}
    lf = {name=Oper12 value=donner}
}
```

Dans l'exemple 1.2, l'entrée du dictionnaire lexical pour EXPOSÉ indique que sa partie du discours profonde est *N* et sa partie du discours de surface est *noun*. L'entrée indique aussi que le troisième actant de EXPOSÉ doit être un nom également et qu'il doit être introduit par la préposition SUR. L'entrée indique aussi les FL qui peuvent s'appliquer et leur valeur [2]. Grâce à ces informations, GenDR peut éviter de produire certaines RSyntP et RSyntS ne respectant pas les restrictions indiquées dans l'entrée de EXPOSÉ, comme par exemple une structure dont le troisième argument de EXPOSÉ aurait une partie du discours autre qu'un nom ou qui serait introduit par la préposition DE.

1.4.3 Les règles

GenDR possède deux modules de règles assurant la transduction des graphes dont nous avons parlé plus haut.

1. Le module sémantique s'occupe de faire passer la RSém à la RSyntP.
2. Le module syntaxique fait passer la RSyntP à la RSyntS.

Chacun des modules comprend des règles permettant de passer d'un niveau de représentation à l'autre ; notamment des règles d'arborisation et de lexicalisation.

Le tableau 1.1 illustre la règle de lexicalisation standard contenue dans le module sémantique de GenDR. La partie gauche s'applique au graphe en entrée et la partie droite contient le graphe de sortie. La partie du bas contient les conditions limitant l'application de la règle,

2. Nous avons présenté ces informations directement dans l'entrée en 1.2, bien qu'en réalité, une partie de celles-ci est définie dans une classe abstraite qui s'applique à toutes les entrées pointant vers elle, ce qui permet de regrouper les entrées partageant des caractéristiques communes.

s'il y a lieu. Dans la règle de lexicalisation standard, le graphe d'entrée est une RSém et le graphe de sortie une RSyntP. Le système cherche une lexicalisation correspondant à un nœud de la RSém dans le DS et l'assigne à une variable. Le système fait ensuite correspondre le nœud de la RSém à un nœud existant dans la RSyntP créé préalablement par une autre règle mais non lexicalisé, et lui assigne la lexicalisation trouvée dans le DS. Le processus est répété pour chaque nœud de la RSém, et de nouvelles structures peuvent être créées tant et aussi longtemps que le système trouve des lexicalisations respectant les conditions de la règle pour un même nœud de la RSém et que ces lexicalisations respectent les contraintes syntaxiques de leurs dépendants et de leurs gouverneurs trouvées dans le dictionnaire lexical.

Sem <=> SyntP Règle de lexicalisation standard	
Partie gauche — Chercher un nœud **Xl** dans la RSém. — Associer une variable **L** à une lexicalisation faisant partie de l'ensemble des lexicalisations associé à **Xl** dans le DS.	**Partie droite** — Chercher un nœud **Xr** déjà créé dans la RSyntP. — **Xr** doit être déjà marqué comme correspondant à **Xl**. — Étiqueter **Xr** avec **L**, telle que définie du côté gauche. — Assigner la valeur de la partie du discours trouvée pour **L** dans le dictionnaire lexical à **Xr**. — Marquer le nœud comme étant lexicalisé au niveau profond.
Conditions — **Xl** a une lexicalisation. — Cette lexicalisation est une entrée dans le dictionnaire lexical. — **Xr** n'est pas déjà lexicalisé. — Si **Xr** a une restriction au niveau de la partie du discours, celle-ci doit être la même que celle de la lexicalisation choisie pour lexicaliser **Xl**.	

TABLEAU 1.1 – Règle de lexicalisation standard dans GenDR

1.4.4 La lexicalisation et l'arborisation

La lexicalisation (Wanner, 1996; Polguère, 1998a, 2000) consiste, pour GenDR, à choisir les bonnes unités lexicales pour exprimer un sémantème. Il existe deux niveaux de lexicalisation dans GenDR :

1. La lexicalisation profonde : lors de la lexicalisation profonde, le système choisit les unités lexicales sémantiquement pleines qui serviront à exprimer les sémantèmes de la RSém. À ce niveau de représentation, GenDR ne s'occupe pas des mots fonctionnels. C'est au problème de la lexicalisation profonde que ce mémoire s'attaque.
2. La lexicalisation de surface : elle fait la correspondance entre les nœuds de la RSyntP et ceux de la RSyntS, notamment en ajoutant à la structure de sortie les unités lexicales fonctionnelles comme les déterminants, les verbes supports et les prépositions requises par le régime des nœuds sémantiquement pleins.

GenDR comprend six types de lexicalisation (Lareau *et al.*, 2018) :

1. La lexicalisation simple pour les lexèmes ;
2. La lexicalisation liée pour les collocations (Lambrey et Lareau, 2015; Lambrey, 2016) ;
3. La lexicalisation par patron pour les locutions (Dubé, 2021) ;
4. La lexicalisation par classe pour les nombres et les noms propres, par exemple ;
5. La lexicalisation de secours pour les mots inconnus ;
6. La lexicalisation grammaticale pour les mots fonctionnels (Galarreta-Piquette, 2018).

Dans cette recherche, nous nous intéressons uniquement à la lexicalisation simple, et laisserons donc les autres types de lexicalisation de côté. Le lectorat peut se référer à Lareau *et al.* (2018) pour une description complète des six types de lexicalisation.

L'arborisation (Polguère, 1990; Galarreta-Piquette, 2018) est étroitement liée à la lexicalisation dans GenDR. Elle consiste d'abord à créer la racine syntaxique de la RSyntP, en la faisant correspondre au sens le plus saillant de la RSém, comme mentionné en §1.4.1. À cette étape, la racine n'est pas encore lexicalisée, et des contraintes y sont appliquées. En français, pour une phrase canonique, on s'assure qu'elle soit un verbe au mode indicatif. Pour ce faire, le système cherche un nœud dans la RSém associé au grammème IND. Ensuite, il cherche dans le DS une lexicalisation dont la partie du discours explicitée dans le dictionnaire lexical est un verbe. Ayant validé ces contraintes, le système copie alors l'étiquette de la lexicalisation sur le nœud correspondant à la racine. Le processus est ensuite répété pour tous les nœuds de la RSém, en respectant leurs contraintes individuelles, du haut vers le bas.

Finalement, GenDR construit les arcs de la RSyntP en les faisant correspondre à ceux de la RSém, à l'aide de règles actancielles recueillant les informations au sujet de la diathèse des nœuds lexicalisés dans le dictionnaire lexical. Nous référons le lectorat à Lareau *et al.* (2018) et Galarreta-Piquette (2018) pour plus de détails au sujet de l'arborisation dans GenDR.

1.5 Synthèse

Nous avons vu que la TST cherche à modéliser la synthèse de la langue (du sens vers le texte) en sept niveaux de représentation linguistique, à l'aide de modules de règles faisant passer chaque niveau de représentation au suivant. En TST, le sens est défini comme le point commun entre un ensemble de paraphrases. Une RSém est donc associée, en théorie, à un nombre potentiellement infini de représentations phonétiques, ou textes. GenDR se veut une application d'un modèle Sens-Texte partiel. En effet, GenDR est basé sur un transducteur de graphes, dont les règles de lexicalisation et d'arborisation et les dictionnaires traduisent une RSém donnée en entrée en RSyntP, et une RSyntP en RSyntS. GenDR se limite donc aux trois niveaux de représentation linguistique les plus profonds de la TST. Une des tâches principales de GenDR, et celle qui nous préoccupe dans la présente recherche, est la lexicalisation profonde. Pour lexicaliser correctement et produire le plus de paraphrases possible pour une même RSém, GenDR doit avoir accès, entre autres choses, à un DS riche qui met en correspondance les sémantèmes de la RSém et les lexicalisations profondes de la RSyntP. Dans le prochain chapitre, nous expliquons comment nous avons construit automatiquement un tel DS compatible avec GenDR.

Chapitre 2

Construction d'un dictionnaire sémantique à partir du Réseau Lexical du Français

2.1 Le Réseau Lexical du Français (RL-fr)

Le RL-fr (Polguère, 2009, 2014) est une ressource linguistique en cours de développement au laboratoire d'Analyse et traitement de la langue française (ATILF). Il s'appuie sur les principes de la Lexicologie Explicative et Combinatoire (LEC) (Mel'čuk *et al.*, 1995; Mel'čuk, 1995; Apresjan, 2000), mais se distingue des ressources lexicographiques traditionnelles par sa structure non linéaire. En effet, le RL-fr se présente sous la forme d'un graphe de type petit-monde (Watts et Strogatz, 1998) et est par définition constitué de nœuds et d'arcs. La plupart de ses nœuds sont des unités lexicales, mais il peut parfois aussi s'agir de phrasèmes (Polguère, 2014). Les arcs, quant à eux, représentent majoritairement des liens lexicaux sémantico-syntaxiques encodés par des FL (Mel'čuk, 1981; Mel'čuk et Polguère, 2021) en TST. Ils peuvent aussi représenter des relations de copolysémie entre les nœuds, des liens sémantiquement vides entre les lexèmes contenus dans un phrasème et le phrasème en question ou encore des liens d'inclusion dans une définition (Polguère, 2014). Dans cette étude, nous ne nous occupons que des nœuds correspondant à des unités lexicales et des arcs correspondant à des FL. Les FL sont centrales à la modélisation du lexique par un réseau lexical. Contrairement à des ressources comme les dictionnaires, qui sont linéaires, ou WordNet (Miller, 1995; Fellbaum, 1998), qui s'appuie majoritairement sur des liens de synonymie et d'hyperonymie, le RL-fr est basé sur une compilation méticuleuse des FL (et

plus rarement des autres relations mentionnées ci-dessus) et par le fait même représente le lexique comme un ensemble d'unités lexicales entretenant des relations de toutes sortes. La figure 2.1 présente un extrait du RL-fr composé uniquement d'unités lexicales et de FL.

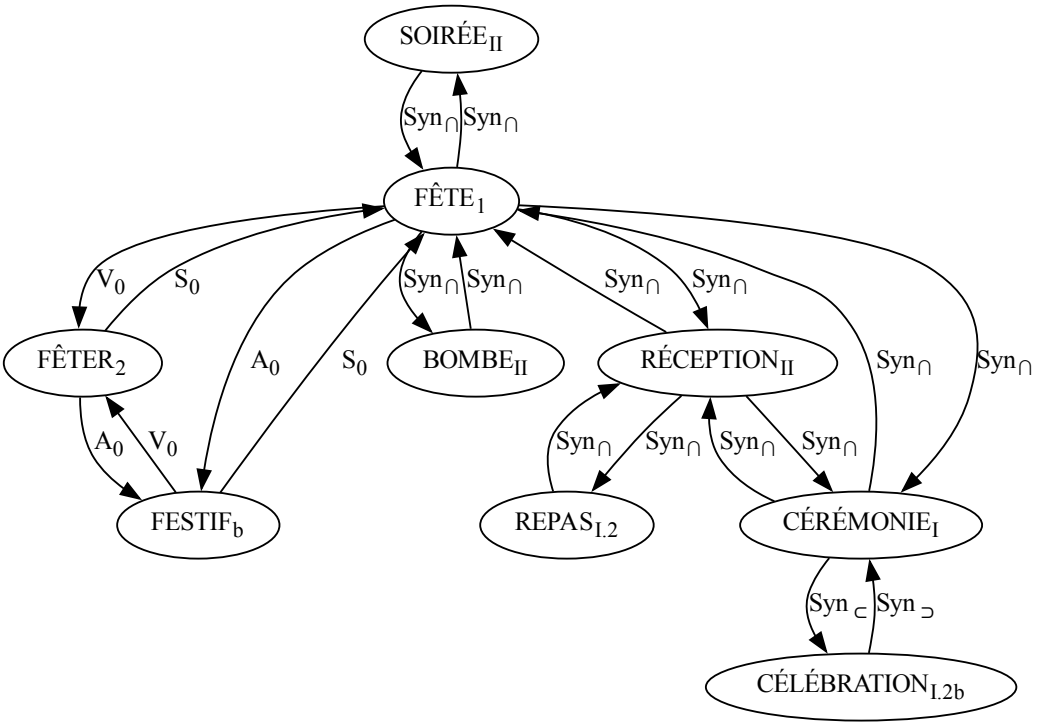


FIGURE 2.1 – Extrait du RL-fr

2.1.1 Les nœuds du RL-fr

Chaque nœud du RL-fr [1] correspond à une unité lexicale du français, ou lexie, et possède une structure interne complexe.

La lexie est l'objet d'étude de la LEC. Selon Mel'čuk (2012), il existe deux types de lexies :

Un lexème est un ensemble de mot-formes qui se distinguent uniquement par la flexion associés à un même sens. Par exemple, CHAT est un lexème.

1. Comme nous ne nous intéressons qu'aux nœuds du RL-fr correspondant aux unités lexicales et que des nœuds d'une autre nature sont assez rares, nous parlerons du RL-fr comme si tous ses nœuds étaient des unités lexicales afin de simplifier le texte.

Une locution est un syntagme figé dont le sens est non compositionnel. Par exemple, DONNER SA LANGUE AU CHAT est une locution.

La lexie possède un sens (le signifié en linguistique saussurienne) (de Saussure et de Mauro, 1994) et une forme graphique ou phonique (le signifiant saussurien). Elle possède également un ensemble de traits de combinatoire, comme le genre, le régime ou la partie du discours. Les traits de combinatoire, en tandem avec les règles grammaticales d'une langue, permettent de combiner correctement les lexies en syntagmes lors de la synthèse du sens vers le texte, et restreignent les textes qu'il est possible de produire à partir d'un sens.

Par exemple, le trait combinatoire indiquant que la lexie LECTURE est un nom empêche la production de la phrase **Papa lecture aux enfants*, mais permet la production de la phrase *Papa fait la lecture aux enfants.*

Ensuite, en LEC, les lexies sont regroupées en vocables. Un vocable est un regroupement de lexies possédant non seulement la même forme mais partageant des liens de sens évidents. Par exemple, le vocable ORANGE regroupe les trois lexies suivantes, tirées du RL-fr :

1. ORANGE$_{\text{I}}$, comme dans *À 3 h du matin, on épluchait les oranges.*
2. ORANGE$_{\text{II.a}}$, comme dans *Un vieil homme, la tête couverte d'un foulard de nylon orange, est assis devant le piano et s'apprête à l'accorder.* (Perec, 1978)
3. ORANGE$_{\text{II.b}}$, comme dans *Le sommet de sa science tient dans la prohibition des couleurs anxiogènes autour du lit ; le rouge et l'orange en particulier* [...] (Pierrat, 2008)

Chaque lexie regroupée sous le vocable ORANGE possède son propre sens, même si elles partagent toutes la même forme. Le vocable permet donc de traiter la polysémie.

Tout comme en LEC, chaque nœud du RL-fr est associé à un seul sens plutôt qu'à un vocable. En effet, les FL ne concernent que les sens, et un nœud du RL-fr en regroupant plusieurs engendrerait une représentation ambigüe du lexique. Le RL-fr est donc un graphe désambiguïsé en ce que chacun de ses nœuds correspond à une seule lexie.

Les nœuds sont étiquetés par un nom normalisé permettant de désambiguïser les différentes lexies regroupées sous un même vocable à l'aide de chiffres romains ou arabes et d'abréviations indiquant la partie du discours si nécessaire, de la manière présentée en 2.1.

Listing 2.1 – Noms normalisés désambiguisant les lexèmes associés au vocable ENTRER

```
entrer#IV.2
entrer#II.2
entrer#IV.1
```

Ensuite, comme mentionné plus haut, chaque nœud possède une structure interne complexe. Lorsqu'on regarde à l'intérieur, on y trouve un grand nombre d'informations lexicographiques, notamment :

— des traits combinatoires, comme la partie du discours ou le genre, s'il y a lieu ;

— la structure actantielle ;

— une définition lexicographique ;

— les FL y étant associées ;

— des exemples lexicographiques.

En bref, les nœuds du RL-fr sont des lexies telles que définies dans la LEC, et chaque nœud possède une structure interne contenant des informations lexicographiques. Maintenant que nous possédons une bonne compréhension de ce que sont les nœuds du RL-fr, nous pouvons passer à la description de son squelette, les relations de FL.

2.1.2 Les fonctions lexicales

En TST, le concept de fonction mathématique est repris pour s'appliquer à la langue. En mathématiques, une fonction est une relation associant chaque élément unique d'un ensemble A à un élément unique d'un ensemble B. Ainsi, une fonction $f(x) = x + 1$ associe chaque valeur de son argument x à sa valeur incrémentée de 1. En TST, les arguments des FL sont plutôt les lexies d'une langue, et les FL les associent à un ensemble de lexies alternatives. Elles servent à encoder des patrons récurrents présents dans les langues. Mel'čuk et Polguère (2021, p. 3) définissent les FL comme suit :

> Une fonction lexicale f appliquée à l'unité lexicale L—ce qui est noté $f(L)$—fournit pour le sens 'σ^f' associé à f un ensemble d'expressions alternatives de ce sens dont la sélection est contrainte par L.

Cela signifie qu'une FL appliquée à une lexie établit la correspondance entre cette lexie et un ensemble d'autres selon le sens de la FL et de la valeur de son argument, si un tel ensemble existe dans la langue.

Par exemple, la FL S_0 prend en argument n'importe quelle lexie et sa valeur, si elle existe dans la langue, est la forme substantive de l'argument.

$$\mathrm{S}_0(\text{COMMENCER}) = \{\text{COMMENCEMENT}\}$$

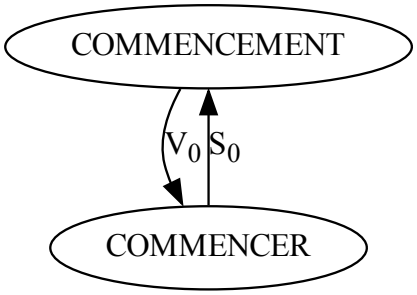


FIGURE 2.2 – Fonctions lexicales S_0 et V_0

Certains nœuds du RL-fr partagent une relation de réciprocité, comme dans le cas de la FL Syn. D'autres partagent une relation inverse. Par exemple, lorsque $\mathrm{V}_0(x) = y$ et que x est un nom, alors $\mathrm{S}_0(y) = x$.

Cette relation est représentée par la figure 2.2.

Toujours selon Mel'čuk et Polguère (2021, p. 13), il existe deux types de fonctions lexicales :

1. Les fonctions paradigmatiques ;
2. Les fonctions syntagmatiques.

Les FL peuvent parfois retourner une valeur de type paradigmatique bien qu'elles soient syntagmatiques et de type syntagmatique bien qu'elles soient paradigmatiques. On dit alors que leur valeur est fusionnée ou défusionnée, respectivement.

Les sections qui suivent décrivent les deux types de FL ainsi que la notion de fusion.

Les fonctions paradigmatiques

Les fonctions paradigmatiques concernent la sélection des lexies par le locuteur ; elles correspondent à des relations de substitution entre celles-ci. Lorsque deux lexies sont liées par une fonction paradigmatique, elles sont en relation de dérivation sémantique. Cela signifie que :

— le sens de la valeur de sortie est inclus dans celui de l'argument ;

— on retrouve l'écart de sens encodé par la FL dans un très grand nombre de paires lexicales ; c'est-à-dire que la FL encode un patron lexical récurrent.

La FL $\mathtt{S}_1$ est un bon exemple d'un tel type de fonction. Sa valeur correspond au substantif du premier actant de l'argument ('celui/ce qui fait ~'). On retrouve un très grand nombre de paires lexicales liées par cette FL.

$$\mathtt{S}_1(\text{POSSÉDER}) = \{\text{PROPRIÉTAIRE}\}$$

$$\mathtt{S}_1(\text{CONSTRUIRE}) = \{\text{CONSTRUCTEUR}, \text{CONSTRUCTRICE}\}$$

$$\mathtt{S}_1(\text{PARTICIPER}) = \{\text{PARTICIPANT}, \text{PARTICIPANTE}\}$$

La relation de dérivation sémantique permet au locuteur de choisir parmi plusieurs lexies pour exprimer un même sens. Avec la FL S_1, on peut exprimer le sens de 'Thomas possède cette maison' soit par *Thomas possède cette maison* ou bien *Thomas est le propriétaire de cette maison*, le sens de 'Thomas construit cette maison' par *Thomas construit cette maison* ou *Thomas est le constructeur de cette maison*, et le sens de 'Thomas participe à cette activité' par *Thomas participe à cette activité* ou *Thomas est un participant de cette activité*.

Les fonctions syntagmatiques

Les fonctions syntagmatiques encodent une relation de collocation entre leur argument et leur valeur. Une collocation est un type de phrasème composé d'une base et d'un collocatif (Mel'čuk et Polguère, 2021). Deux lexies sont en relation de collocation lorsque :

— la lexie formant la base est sélectionnée par le locuteur indépendemment du collocatif ;

— le collocatif est sélectionné pour exprimer le sens de la FL le reliant à la base en fonction de cette dernière.

L'argument d'une FL syntagmatique correspond à la base de la collocation, et la valeur au collocatif. Un bon exemple de FL syntagmatique est `Magn`. Le sens de `Magn` exprime l'intensification. Ainsi, sa valeur de sortie sera en relation de collocation avec l'argument et exprimera son intensification.

$$\mathtt{Magn}(\text{PEUR}) = \{\text{BLEU}, \text{PANIQUE}\}$$

$$\mathtt{Magn}(\text{ÉCHEC}) = \{\text{CUISANT}\}$$

$$\mathtt{Magn}(\text{PLUIE}) = \{\text{BATTANT}, \text{DILUVIEN}\}$$

Les fonctions à valeur fusionnée ou défusionnée

Il arrive parfois que la valeur d'une FL syntagmatique soit fusionnée, c'est-à-dire qu'elle soit de type paradigmatique plutôt que syntagmatique, ou que la valeur d'une FL paradigmatique soit défusionnée, c'est-à-dire qu'elle soit de type syntagmatique. La FL paradigmatique A_1, qui exprime le qualificatif adjectival typique du premier actant sémantique de l'argument, en est un bon exemple.

$$\mathrm{A}_1(\text{AMOUR}) = \{\text{AMOUREUX}, \textit{en} \sim\}$$

$$\mathrm{A}_1(\text{ADMIRATION}) = \{\text{ADMIRATIF}, \textit{en} \sim\}$$

L'ensemble retourné par la fonction A_1 pour AMOUR et ADMIRATION, contient à la fois :

— une valeur paradigmatique (AMOUREUX, ADMIRATIF) en relation de substitution avec l'argument. En effet, on peut exprimer le sens 'Thomas ressent de l'amour/admiration pour Trinity' par *Thomas est amoureux/admiratif de Trinity*.

— une valeur syntagmatique (EN ~) qui correspond au collocatif de l'argument. On peut en effet ajouter EN devant l'argument et dire *Thomas est en amour avec/admiration devant Trinity*[2] pour exprimer 'Thomas ressent de l'amour/admiration pour Trinity'.

2.2 Méthodologie

Dans un DS, une entrée est un sémantème et sa valeur l'ensemble des lexicalisations correspondantes dans une langue. Les nœuds du RL-fr, étant des lexies, correspondent à un seul sémantème et sont donc désambiguïsés (voir §2.1).

Comme il faut bien avoir une manière de nommer ces sémantèmes, le nom normalisé (§2.1) d'un nœud du RL-fr peut aussi servir à identifier son sens, et peut donc faire office de sémantème. Ainsi, nous avons pu faire du nom normalisé de chaque nœud du RL-fr une entrée du DS. Les lexicalisations associées correspondent, quant à elles, aux nœuds liés à l'entrée par des FL spécifiques dans le RL-fr. Un ensemble de lexicalisations associé à un sémantème dans le DS regroupe les lexies partageant un même sens ; ce ne sont donc pas toutes les FL du RL-fr liées à un nœud qui retournent une valeur l'exprimant. Nous devions d'abord faire un tri pour ne garder que les FL pertinentes pour nous. Comme les FL paradigmatiques retournent des unités lexicales en relation de dérivation sémantique avec l'argument et dont le

2. La sélection des dépendants syntaxiques de la valeur de sortie d'une FL peut varier mais est traitée plus loin dans la synthèse sens-texte. Elle n'a donc pas d'impact sur les valeurs possibles d'une FL.

sens est inclus dans celui de l'argument (§2.1.2), c'est ce type de FL qui nous intéressait, plus particulièrement une famille de FL que nous appelons les fonctions lexicales paradigmatiques sémantiquement vides.

Les fonctions lexicales paradigmatiques sémantiquement vides (FLPSV) sont des FL paradigmatiques qui n'apportent ni n'enlèvent de sens à leur argument. On peut concevoir la valeur des FLPSV comme un synonyme de leur argument mais non limité à la partie du discours de ce dernier. Les FL suivantes donnent un aperçu de ce que sont les FLPSV :

— Syn(MOURIR) = ⌜CASSER SA PIPE⌝

— A_4(NETTOYER) = NETTOYANT

— S_0(PRÉSENTER) = PRÉSENTATION

— A_0(DÉSESPOIR) = DÉSÉSPÉRÉ

— V_0(TRAHISON) = TRAHIR

— Adv_1(INDISCRÉTION) = INDISCRÈTEMENT

Les FLPSV peuvent être approximatives ou exactes. En effet, certaines FLPSV retournent une valeur dont le sens est approximatif par rapport à celui de l'argument, tout en restant très semblable. C'est cette propriété qui nous permettra d'élaborer le paramètre permettant de régler la distance sémantique maximale entre les entrées et leurs lexicalisations.

En ajoutant les nœuds reliés par des FLPSV à l'ensemble de lexicalisations associé à une entrée du DS, on obtient des résultats tels que présentés dans le listing 2.2, où des lexicalisations sémantiquement équivalentes entre elles et à l'entrée ne partagent pas la même partie du discours.

Cela ne pose pas problème pour GenDR, car comme nous l'avons expliqué dans la §1.4, des restrictions dans les règles sur les nœuds des structures d'entrée et de sortie empêchent le système de produire des RSyntP agrammaticales et combinent la lexie choisie pour lexicaliser un sémantème et les autres éléments de la RSyntP de façon adéquate.

Voici comment nous avons procédé pour extraire les FLPSV du RL-fr :

1. Nous avons compilé dans un document les noms des FLPSV tels que présentés dans le RL-fr. Pour rendre cette liste la plus générale possible, nous avons écrit le nom des FLPSV sous forme d'expressions régulières. Pour déterminer quelles FL étaient les FLPSV que nous voulions attraper avec les expressions régulières, nous nous sommes appuyés sur les travaux de Mel'čuk et Polguère (2021).

Listing 2.2 – Lexicalisations ne partageant pas la même partie du discours

```
"Espagnol_N#I" {
    lex="Espagne"
    lex="Espagnol_N#I"
    lex="Espagnole"
    lex="espagnol_Adj#I.1"
    lex="espagnol_Adj#I.2"
}
"Satan" {
    lex="Satan"
    lex="diable#I.a"
    lex="diabolique"
    lex="diaboliquement"
}
```

2. Pour chaque expression régulière, nous avons indiqué par 0 ou 1 si la valeur de sortie des FLPSV correspondantes est une lexicalisation dont le sens correspond approximativement ou exactement à celui de l'argument. En effet, la valeur de sortie de certaines FLPSV est une lexicalisation exacte (0) de son argument, et la valeur d'autres correspond à une lexicalisation dont le sens est seulement quasi-identique à celui de son argument, une quasi-lexicalisation (1). Comme nous élaborons un module de lexicalisation souple, la distinction entre les lexicalisations exactes et les quasi-lexicalisations est cruciale pour pouvoir concevoir le paramètre permettant de décider du degré d'exactitude des lexicalisations par rapport à leur entrée (voir §1.1).
3. Nous avons également indiqué si la valeur de sortie de la FLPSV possède la même partie du discours que l'argument. Cette information sera importante lors de l'évaluation des lexicalisations supplémentaires générées par un modèle de langue neuronal (voir chapitre 4).
4. Ensuite, nous avons comparé les expressions régulières avec les noms des FL du RL-fr et en avons extrait celles qui y correspondaient. Nous avons également extrait les informations concernant le type de ces FLPSV (syntagmatique ou paradigmatique). Les FLPSV sont essentiellement, comme leur nom l'indique, des FL paradigmatiques. Toutefois, certaines se sont avérées être des FL syntagmatiques en usage fusionné (§2.1.2).

Les tableaux 2.1 et 2.2 présentent un aperçu des données obtenues en suivant les étapes que nous décrivons ci-dessus. La liste complète des expressions régulières annotées et la liste complète des FLPSV récupérées grâce à celles-ci se trouvent dans l'annexe A, respectivement

dans les tableaux A.1 et A.2.

Expression régulière	Approximatif	Même PDD
`^A_\d$`	0	0
`^S_0Pred$`	0	0
`^Syn_[∩⊂⊃]$`	1	1
`^Adv_\d$`	0	0

TABLEAU 2.1 – Quelques patrons de FLPSV à extraire du RL-fr

ID	Nom	Type	Approximatif	Même PDD
`ls:fr:lf:627`	`A_4`	paradigmatic	0	0
`ls:fr:lf:366`	`S_0Pred`	paradigmatic	0	0
`ls:fr:lf:4`	`Syn_⊂`	paradigmatic	1	1
`ls:fr:lf:161`	`Adv_1`	syntagmatic	0	0

TABLEAU 2.2 – Quelques FLPSV trouvées dans le RL-fr à l'aide d'expressions régulières

Armée d'une table contenant les FLPSV et leurs attributs, nous pouvions nous lancer dans la construction du DS. Une version de base du script permettant de construire un DS à partir de quelques FL existait déjà, et nous l'avons adaptée pour obtenir les résultats désirés, en y ajoutant notamment un paramètre d'approximation maximale (PAM). Le PAM est appliqué au script construisant le DS et sa valeur doit être un entier. Il permet de décider du degré d'approximation maximal des lexicalisations par rapport à l'entrée associée dans le DS produit par le module de lexicalisation. Dans les étapes ci-dessous, nous décrivons le fonctionnement du script et du PAM.

1. Nous avons commencé par construire un dictionnaire Python avec comme entrées les sémantèmes correspondant aux nœuds du RL-fr, et comme valeurs des ensembles vides. Pour ce faire, nous avons simplement extrait le nom normalisé de chaque nœud et en avons fait une entrée du dictionnaire.
2. Nous avons ajouté la lexicalisation triviale de chaque entrée. Comme mentionné plus haut (§2.1), chaque nœud du RL-fr porte une étiquette servant à l'identifier : son nom normalisé. La lexicalisation triviale consiste simplement à recopier ce nom normalisé et en faire une première lexicalisation. En effet, comme les entrées du DS, qui sont des sémantèmes, sont étiquetées par le nom normalisé du nœud du RL-fr au sens correspondant, elles peuvent toujours au moins être exprimées dans la langue par cette étiquette.

Par exemple, le sens étiqueté ‘manger$_{I.1a}$’ peut toujours être lexicalisé par MANGER$_{I.1a}$, et il en va de même pour toutes les entrées du DS.

3. Nous avons extrait du RL-fr et ajouté à l'ensemble des lexicalisations d'une entrée tous les nœuds y étant liés directement par une FLPSV et dont la valeur d'approximation ne dépasse pas la valeur du PAM.

4. Le RL-fr indique si les FL sont fusionnées. Grâce à cette information, nous avons pu ignorer les valeurs de sortie des FL paradigmatiques en usage syntagmatique et inclure celles des FL syntagmatiques en usage paradigmatique. Le tableau 2.3 présente un extrait des données du RL-fr qui nous ont permis d'ajouter à l'ensemble de lexicalisations de chaque entrée la lexie cible (la valeur de sortie) directement liée par une FLPSV à la source (l'argument) équivalente à l'entrée. Nous nous sommes assurée de la réciprocité de chaque FLPSV en inversant la source et la cible et en ajoutant cette dernière à l'ensemble des lexicalisations de l'entrée équivalente à la source.

5. Nous avons ajouté récursivement les lexicalisations liées aux lexicalisations directement liées à l'entrée en limitant la distance sémantique entre les lexicalisations extraites et l'entrée associée avec le PAM.

   — La figure 2.3 présente un extrait du RL-fr avec, à la place des noms des FLPSV reliant les nœuds, leur valeur d'approximation telle que nous l'avons annotée lors de leur extraction du RL-fr (voir page 33). Pour s'assurer que le degré d'approximation d'une lexicalisation respecte le PAM et que celle-ci peut être ajoutée à l'ensemble associé à une entrée, un compteur est initialisé à 0. Pour chaque lexicalisation parcourue dans le RL-fr à partir de l'entrée, le script ajoute la valeur d'approximation de la ou des FLPSV la reliant à l'entrée à ce compteur. À chaque nœud visité, la valeur du compteur est comparée à celle du PAM. Si la somme dépasse la valeur du PAM, la lexicalisation est ignorée et n'est pas ajoutée à l'ensemble des lexicalisations de l'entrée. L'exploration du RL-fr est alors interrompue et le script revient à son point d'entrée pour explorer un autre chemin. Ainsi, lorsque PAM = 0, le script n'ajoute à l'ensemble des lexicalisations d'une entrée que les lexicalisations exactes, et lorsque PAM > 0, il peut y ajouter des quasi-lexicalisations. Dans la figure 2.4, les lexicalisations en vert de l'entrée ‘fête’ en rouge peuvent être ajoutées à l'ensemble des lexicalisations lorsque PAM = 0, les lexicalisations en bleu peuvent uniquement y être ajoutées lorsque PAM = 1 et les lexicalisations en violet uniquement lorsque PAM > 1.

— Les lexicalisations dont la somme de la valeur d'approximation des FL les reliant à l'entrée est inférieure au PAM sont toujours ajoutées à l'ensemble.

— Le listing 2.4 présente du pseudocode représentant le script utilisé pour extraire récursivement les lexicalisations du RL-fr à partir des lexicalisations déjà compilées tout en respectant la valeur du PAM. La variable `DS` correspond au dictionnaire Python construit aux étapes 1 à 4, dont chaque entrée est associée à l'ensemble de paires `(lexicalisation, distance sémantique)` correspondant aux lexicalisations et à leur distance sémantique avec l'entrée leur étant directement liée dans le RL-fr. Le paramètre `PAM` correspond à la valeur du PAM. La variable `DS` possède la structure présentée dans le listing 2.3.

6. Finalement, nous écrivons le dictionnaire Python dans un fichier texte qui sera intégré à GenDR et qui se présente sous la forme présentée dans le listing 2.5.

Source	**FL**	**Cible**	**Type**	**Fusion**	**Approx.**	**Même PDD**
EXPLIQUER	A_4	EXPLICATIF	Paradigm.	0	0	0
AGILE	S_0Pred	AGILITÉ	Paradigm.	0	0	0
ADIEU	Syn_c	AU REVOIR	Paradigm.	0	1	1
AFFECTION	Adv_1	AFFECTUEUSEMENT	Syntagm. [a]	1	0	0

a. Le type de certaines fonctions autrefois syntagmatiques, notamment Adv_i, a été mis à jour et est maintenant paradigmatique. Toutefois, les données de RL-fr n'ont pas encore été mises à jour en conséquence, ce qui explique pourquoi la fonction Adv_1 est de type syntagmatique dans le tableau.

TABLEAU 2.3 – Quelques liens trouvés dans le RL-fr avec les FLPSV

Listing 2.3 – Structure de la variable *DS* dans le listing 2.4

```
"admiration": {("admiration", 0), ("admirer#I", 0), ("admiratif",
   0)}
"adresse courriel": {("mél#II.1", 0), ("mail#II.1", 0), ("adresse
   e-mail", 0), ("adresse mail", 0), ("e-mail#II.1", 0), ("adresse
   courriel", 0), ("adresse de courrier électronique", 0),
   ("adresse électronique", 0)}
```

On remarque dans la figure 2.4 que plusieurs chemins à partir de l'entrée peuvent être parcourus pour arriver à une même lexicalisation. L'algorithme que nous avons élaboré ne repasse pas par un chemin qu'il a déjà parcouru grâce à une liste qui garde en mémoire les nœuds visités, afin d'éviter le parcours du graphe à l'infini. Comme notre algorithme ne

Listing 2.4 – Pseudocode du script d'extraction récursive des lexicalisations du RL-fr

```
def expand_lexs(DS, PAM)
    pour chaque entrée : ensemble de lexicalisations du DS :
        lexs récursives = get_lexs(DS, entrée, PAM, compteur de
           départ=0, visités=liste vide)
        ajouter les lexs récursives à l'ensemble de lexicalisations
    retourner le DS

def get_lexs(DS, entrée, PAM, compteur de départ, visités)
    ajouter entrée à la liste visités # ne pas parcourir plus
       d'une fois le même chemin
    lexs récursives = ensemble vide
    ensemble de lexicalisations = ensemble associé à entrée dans
       le DS
    pour chaque paire (lexicalisation, distance sémantique) dans
       l'ensemble de lexicalisations:
        distance sémantique totale = distance sémantique +
           compteur de départ
        si la lexicalisation n'est pas dans la liste visités et
           que la distance sémantique totale <= PAM :
            nouvelle lex = (lexicalisation, distance sémantique
               totale) # connaître la distance sémantique
            ajouter la nouvelle lex aux lexs récursives
            ajouter aux lexs récursive le résultat de get_lexs(DS,
               lexicalisation, PAM, distance sémantique totale,
               visités))
    retourner les lexs récursives
```

parcourt pas nécessairement le chemin le plus court en premier, cette particularité constitue une limite. En effet, les nœuds ajoutés dépendent de l'ordre dans lequel le script parcourt le réseau, et si le chemin le plus court n'est pas visité en premier, certains nœuds risquent d'être ignorés même dans le cas où il existe un chemin assez court pour respecter la valeur du PAM.

Le PAM a aussi pour but de rendre la lexicalisation plus flexible pour un utilisateur. En effet, quelqu'un utilisant GenDR pourrait avoir besoin d'un texte correspondant parfaitement à la RSém d'entrée, et quelqu'un d'autre pourrait avoir besoin de textes moins précis mais de plus de paraphrases. Le paramètre intègre donc cette souplesse à notre module de lexicalisation.

Listing 2.5 – Exemple d'une entrée du DS avec PAM

```
"incorporer#II" {
  lex="incorporation#II"
  lex="incorporer#II"
  qlex="intégration#I"
  qlex="intégration#II.1"
  qlex="intégrer#I"
  qlex="s'intégrer#I"
  qlex="s'intégrer#II.1"
}
```

Le module de lexicalisation souple complet se trouve dans l'annexe B, dans le listing B.1.

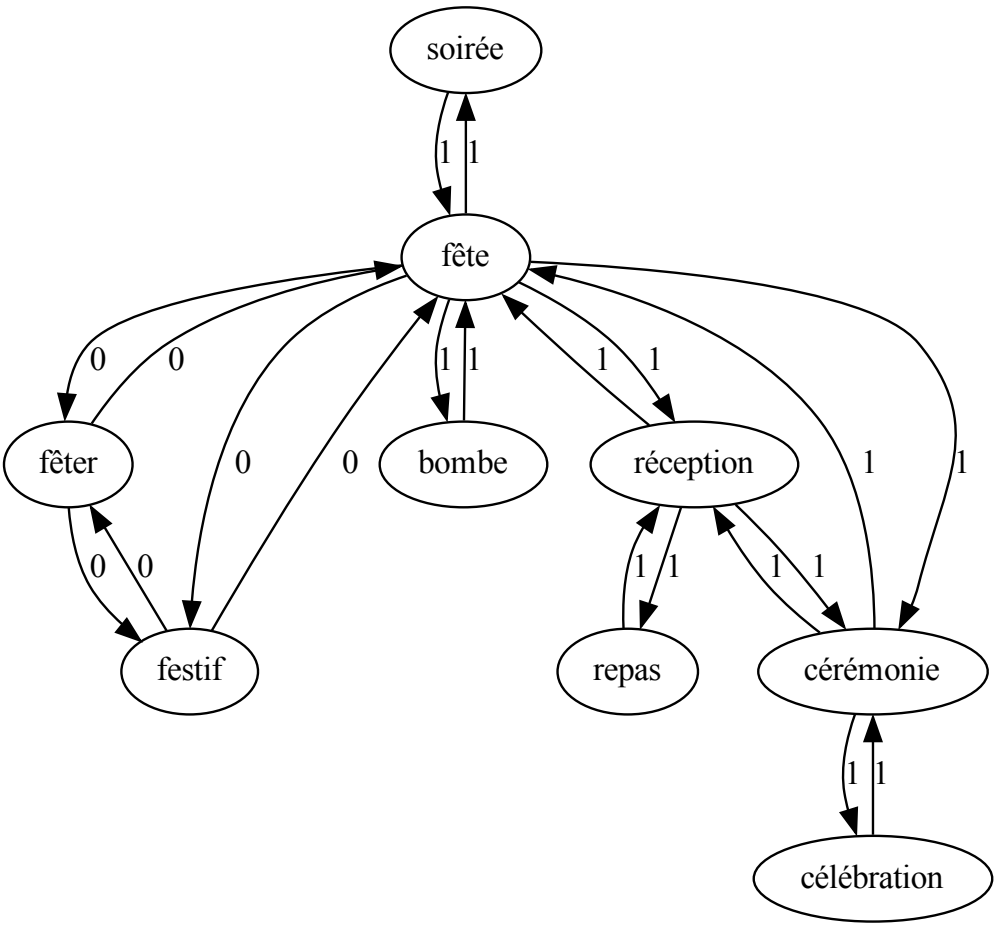


FIGURE 2.3 – Extrait du RL-fr présentant la valeur d'approximation des FLPSV

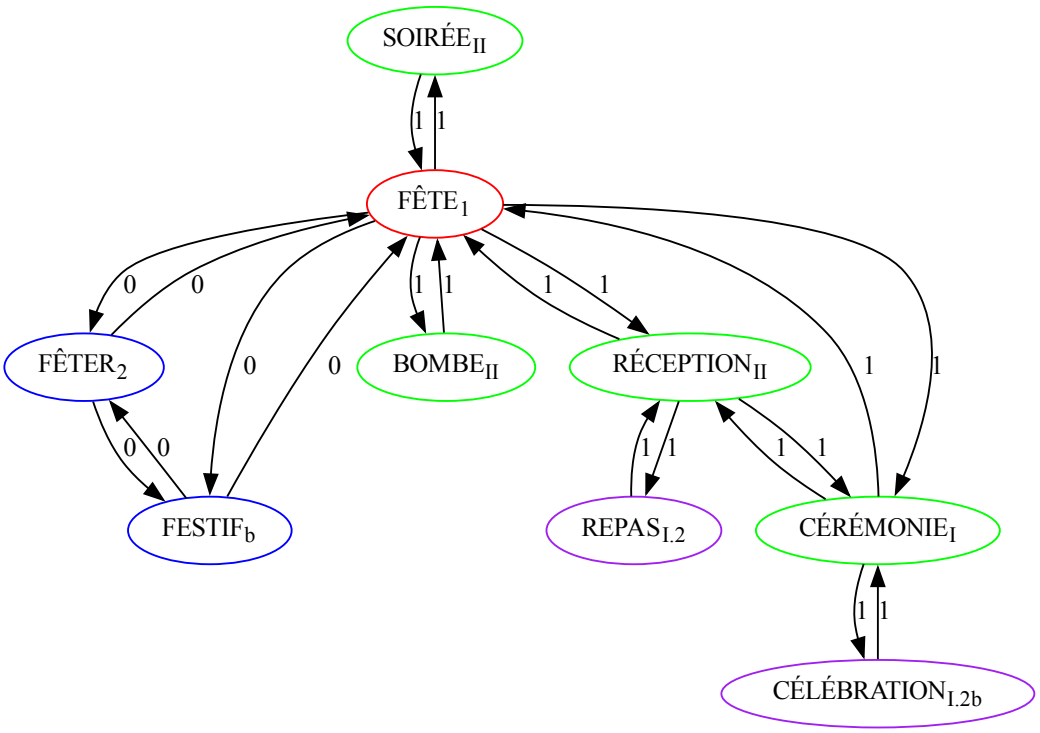


FIGURE 2.4 – Lexicalisations retenues lorsque PAM = 0 (en bleu), PAM = 1 (en vert et en bleu) et PAM = 2 (en bleu, en vert et en violet)

2.3 Évaluation

Le DS produit à l'aide du RL-fr est compatible avec GenDR. À partir des 29 399 nœuds du RL-fr, nous avons créé le même nombre d'entrées dans le DS. Chacune d'entre elle est associée à ses lexicalisations par un certain nombres de liens, variable selon la distance sémantique maximale choisie par l'utilisateur. Lorsque PAM = 0, les entrées du DS sont associées à leurs lexicalisations par 49 235 liens de lexicalisation. Ce nombre augmente avec la distance sémantique maximale, et atteint 572 686 liens de lexicalisation lorsque PAM = 5. Nous avons donc fait passer le DS contenu dans GenDR de 1 425 entrées associées à 1 555 lexicalisations à un DS de 29 399 entrées et 29 399 lexicalisations uniques reliées entre elles par un minimum de 49 235 liens. La capacité de GenDR à produire des paraphrases s'en voit ainsi décuplée.

Toutefois, notre algorithme de parcours de graphe pourrait être amélioré. En effet, s'il existe plusieurs chemins pour atteindre un même nœud à partir d'une même entrée, tout dépendant de l'ordre dans lequel le script visite le graphe, certains nœuds sont ignorés, même s'ils répondent au critère du PAM. Un algorithme calculant le plus court chemin entre un nœud et son entrée tout en respectant la valeur du PAM aurait été plus approprié. Avec un tel algorithme, on peut présumer que le nombre de lexicalisations associées à chaque sémantème du DS aurait été encore plus élevé, puisqu'à cause de la structure de notre algorithme, certaines lexicalisations pertinentes ont été oubliées.

Dans le réseau simplifié de la figure 2.5, on peut voir qu'avec PAM = 2, l'algorithme partant du nœud A et passant d'abord par B pour atteindre C ne peut ajouter le nœud D à l'ensemble des lexicalisations de A, puisque ce chemin dépasse la valeur du PAM. Bien qu'il existe un autre chemin entre A et D ne la dépassant pas, après avoir visité le premier chemin, l'algorithme étiquette le nœud C comme étant déjà visité, et ne peut donc pas aller plus loin une seconde fois. Le nœud D est donc ignoré, malgré le fait qu'au moins un chemin entre A et D respecte la valeur du PAM. Un algorithme idéal choisirait donc le plus court chemin entre un nœud et sa source en priorité.

Malgré ces nœuds ignorés, la couverture du DS produit a été considérablement élargie.

Nous présentons dans la figure 2.7 un certain nombre de RSyntP produites par GenDR pour la RSém de la figure 2.6 à l'aide du DS produit par notre algorithme. La sous-figure 2.7a correspond à sa lexicalisation triviale. On voit qu'un grand nombre de synonymes associés à une entrée permet de produire une base solide de paraphrases, et que des lexicalisations d'autres parties du discours, avec les règles et les entrées du dictionnaire lexical appropriées, peuvent être combinées pour produire des paraphrases aux structures syntaxiques variées. Le

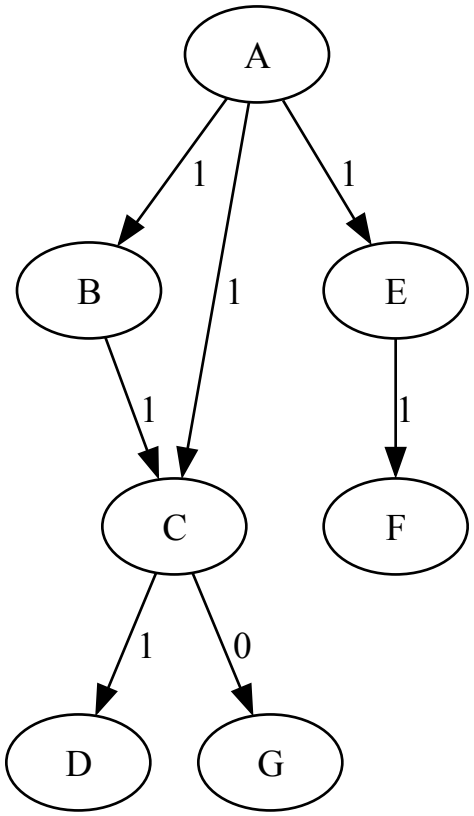


FIGURE 2.5 – Réseau exemple

premier objectif de la présente recherche, qui consistait à construire automatiquement un DS riche à partir du RL-fr, a donc été atteint.

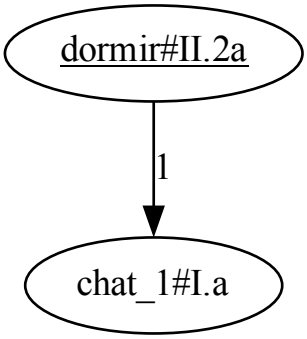
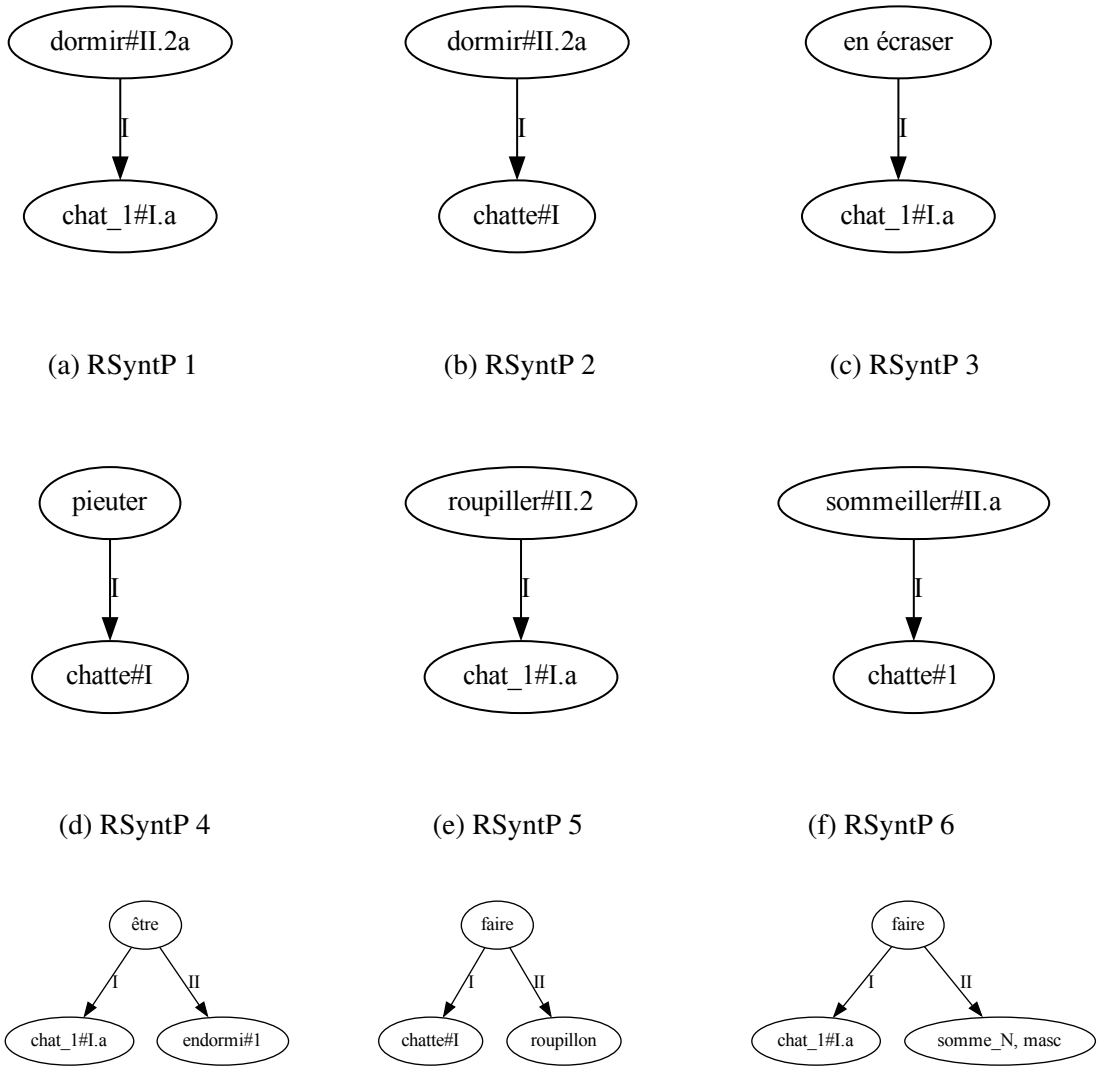


FIGURE 2.6 – Représentation sémantique



(a) RSyntP 1 (b) RSyntP 2 (c) RSyntP 3

(d) RSyntP 4 (e) RSyntP 5 (f) RSyntP 6

(g) RSyntP 7 (h) RSyntP 8 (i) RSyntP 9

FIGURE 2.7 – RSyntP produites par GenDR

2.4 Synthèse

Dans ce chapitre, nous avons présenté brièvement le RL-fr, qui modélise le lexique du français sous forme de graphe. Celui-ci est composé de nœuds et d'arcs, qui représentent respectivement en majorité les lexies du français et des FL encodant les patrons récurrents dont elles font partie. Les nœuds ont une structure interne riche qui contient plusieurs informations lexicographiques, et les FL sont de deux types : paradigmatique et syntagmatique. Les FL paradigmatiques encodent une relation de dérivation sémantique, et les FL syntagmatiques une relation de collocation. Les FL peuvent aussi être fusionnées ou défusionnées, c'est-à-dire qu'une FL syntagmatique peut retourner une valeur paradigmatique et une FL paradigmatique une valeur syntagmatique, respectivement.

Le RL-fr est une ressource toute indiquée pour construire un DS. En effet, il est composé de lexies, chacune associée à un seul sémantème, et les données contenues dans un DS sont justement des sémantèmes associés à des lexies.

Pour construire le DS, nous avons conçu un module de lexicalisation souple qui réorganise les données du RL-fr sous forme de dictionnaire en faisant du nom normalisé de chaque nœud une entrée et en faisant des nœuds liés à ceux correspondant à chaque entrée par des FLPSV ses lexicalisations. Nous avons déterminé si ces FLPSV retournaient une valeur exacte ou approximative afin d'élaborer le PAM, qui permet de régler la distance sémantique maximale qui sépare les lexicalisations de leur entrée.

Le résultat obtenu est un DS au degré d'approximation variable, compatible avec GenDR et qui contient 29 399 entrées liées entre elles par un minimum de 49 235 liens, lorsque PAM = 0. Le nombre de liens de lexicalisation augmente avec la valeur du PAM.

Dans le prochain chapitre, nous présentons les plongements lexicaux et explorons la possibilité d'utiliser un modèle de langue neuronal pour augmenter la couverture du RL-fr.

Chapitre 3

Utilisation d'un modèle de langue neuronal pour enrichir le dictionnaire sémantique

3.1 Les plongements lexicaux

Les plongements lexicaux, souvent appelés vecteurs sémantiques, se veulent une représentation mathématique du sens des mots qui permettent à l'ordinateur de « comprendre » les langues naturelles. Les sections qui suivent décrivent brièvement l'origine et le fonctionnement des plongements lexicaux, ainsi que le modèle vectoriel contextuel que nous avons testé pour générer des lexicalisations supplémentaires pour le DS.

3.1.1 L'hypothèse distributionnelle

Le fonctionnement des plongements lexicaux se base sur l'hypothèse distributionnelle (Harris, 1954; Firth, 1957), qui postule que les mots apparaissant dans des contextes similaires ont tendance à avoir des sens similaires. Par exemple, dans un corpus, des tokens qui apparaissent régulièrement dans le contexte de *manger* ont tendance à avoir un sens relié à la nourriture. Pour connaître les contextes dans lesquels un token apparaît dans un corpus, on le compare avec tous les autres tokens du corpus et on vérifie en compagnie desquels il apparaît, et à quelle fréquence. On obtient alors une liste de nombres de la longueur du vocabulaire du corpus ; le vecteur du token. Si on répète cette opération avec tous les tokens du corpus, on obtient une matrice de co-occurrences de taille $N \times N$, où N est la taille du vocabulaire.

Par exemple, considérons un corpus composé des phrases en (1) :

(1) a. Le chien aboie.
b. Le chat miaule.
c. Le bébé babille.

Après avoir enlevé les tokens vides sémantiquement, on obtient un vocabulaire composé des six types suivants : {*chien*, *aboie*, *chat*, *miaule*, *bébé*, *babille*}

La matrice de co-occurrences met en relation tous les types du vocabulaire les uns avec les autres et compte le nombre de fois où chacun d'entre eux apparaît avec les autres, comme présenté dans le tableau 3.1.

	chat	chien	bébé	aboie	miaule	babille
chat	1	0	0	0	1	0
chien	0	1	0	1	0	0
bébé	0	0	1	0	0	1
aboie	0	1	0	1	0	0
miaule	1	0	0	0	1	0
babille	0	0	1	0	0	1

TABLEAU 3.1 – Matrice de co-occurrences

Le vecteur de chaque types du vocabulaire correspond à la ligne de la matrice de co-occurrences lui étant associée, et est de taille $1 \times N$. Dans notre exemple, les vecteurs sont donc les suivants :

chat : $[1, 0, 0, 0, 1, 0]$

chien : $[0, 1, 0, 1, 0, 0]$

bébé : $[0, 0, 1, 0, 0, 1]$

aboie : $[0, 1, 0, 1, 0, 0]$

miaule : $[1, 0, 0, 0, 1, 0]$

babille : $[0, 0, 1, 0, 0, 1]$

On peut ensuite placer les vecteurs dans un espace vectoriel à N dimensions. Comme un espace à autant de dimensions est difficile à visualiser, on peut appliquer des techniques de réduction des dimensions comme la décomposition en valeurs singulières (Carroll et Chang, 1970; Stewart, 1993; Abdi, 2007), l'analyse en composantes principales (Pearson, 1901; Hotelling, 1933) ou l'algorithme t-SNE (van der Maaten et Hinton, 2008) pour en faire un espace

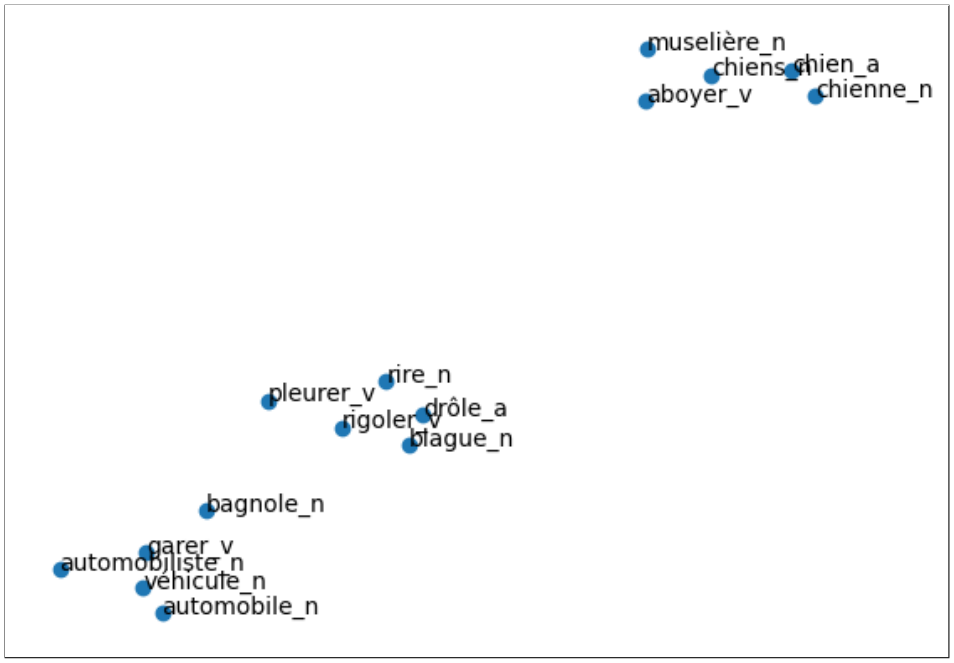


FIGURE 3.1 – Vecteurs calculés par Word2Vec

à deux ou trois dimensions et visualiser la proximité ou l'éloignement des vecteurs correspondants aux types du vocabulaire les uns par rapport aux autres. On trouve que, conformément à l'hypothèse distributionnelle, les types se rapprochant dans l'espace vectoriel ont tendance à avoir des sens similaires, et les types éloignés à avoir des sens différents. La figure 3.1 générée avec l'alorithme t-SNE en fait l'illustration.

Notre corpus jouet indique que les tokens *chat* et *miaule* apparaissent dans le même contexte, *chien* et *aboie* dans un autre et *bébé* et *babille* encore dans un autre. Un modèle construit à partir de ces données présenterait les tokens composant chaque paire comme fortement liés au plan sémantique. En réalité, les modèles vectoriels modélisent la langue à partir de corpus contenant des millions, voire des milliards de mots et les résultats obtenus sont beaucoup plus fins. Tous sont toutefois basés sur l'hypothèse distributionnelle.

3.1.2 Word2Vec

Word2Vec (Mikolov *et al.*, 2013) est un réseau neuronal superficiel ayant révolutionné la sémantique distributionnelle. Il s'agit d'un modèle vectoriel de type *sac de mots*, c'est-à-dire qu'il encode des coordonnées statiques pour chaque type rencontré dans le vocabulaire à l'entraînement et les garde en mémoire. Word2Vec produit de très bons résultats, et l'une des démonstrations les plus célèbres de l'efficacité du modèle est l'équation $\overrightarrow{king} - \overrightarrow{man} +$

$\overrightarrow{woman}$ = $\overrightarrow{queen}$. Bien qu'on puisse atteindre ce résultat en effectuant cette opération sur un modèle pré-entraîné de Word2Vec, celui-ci n'est pas sans présenter ses limites.

Tout d'abord, Word2Vec est incapable de calculer le vecteur d'un mot qu'il n'a pas vu à l'entraînement. Cela représente une limite car l'entraînement est déjà long et les corpus, aussi étendus soient-ils, ne contiennent jamais tous les tokens d'une langue. Ainsi, nous courrions le risque que Word2Vec ne connaisse pas la lexie pour laquelle nous cherchions les plus proches voisins, et soit incapable de nous proposer quoi que ce soit.

Ensuite, Word2Vec fait fi de l'ambiguïté lexicale présente dans toutes les langues. En effet, beaucoup de vocables regroupent plus d'un sens, mais Word2Vec encode les coordonnées vectorielles des types. Le modèle calcule le sens d'un type en faisant une moyenne de tous les vecteurs calculés pour chaque occurrence de ce type dans le corpus. Ainsi, les sens les plus communs sont privilégiés par rapport aux sens plus rares, et le produit final est un vecteur quelque peu approximatif qui essaie de représenter tous les sens du type en même temps.

Ainsi, en ce qui a trait à notre recherche, nous ne pouvions pas utiliser Word2Vec ou tout autre modèle de type *sac de mots* car les nœuds du RL-fr sont désambiguïsés. En effet, ils correspondent à des lexies, qui sont par définition associées à un seul sémantème. Les vecteurs de Word2Vec, eux, correspondent plutôt aux vocables polysémiques regroupant les lexies, et parfois même à plusieurs vocables ayant la même forme, dans le cas de l'homonymie.

Le nom normalisé des nœuds du RL-fr permet de désambiguïser les différentes lexies regroupées sous un même vocable et partageant donc une même forme à l'aide d'un code composé de chiffres et de lettres (voir §2.1.1). Le nom normalisé ne pouvait donc pas être un mot reconnu par Word2Vec. En enlevant ces codes des noms normalisés, il devient impossible de distinguer les différentes lexies partageant une même forme, pour nous comme pour Word2Vec. Ce dernier aurait retourné les tokens ayant le vecteur le plus similaire à un vecteur correspondant à une moyenne peu précise des sens qu'il aurait calculés pour toutes les occurrences d'un vocable. Les tokens proposés par Word2Vec pour enrichir les entrées du DS, elles-mêmes ne correspondant qu'à un seul sémantème chacune, seraient donc les plus proches voisins dans l'espace vectoriel d'un vecteur regroupant lui-même plusieurs sens.

Nous nous sommes donc tournée vers les modèles vectoriels contextuels et avons choisi de travailler avec camemBERT (Martin *et al.*, 2020), une version de BERT (Devlin *et al.*, 2019) entraînée sur un corpus en français.

3.1.3 BERT

BERT (*Bi-directional Transformers for Language Understanding*) est un modèle vectoriel de type *Transformers* (Vaswani *et al.*, 2017) qui calcule le vecteur de chaque token d'une séquence en fonction de l'utilité des autres tokens de la séquence pour le calcul du token en question. Ces vecteurs contextuels se veulent une solution au problème de l'ambiguïté sémantique auquel font face Word2Vec et les autres modèles encodant des vecteurs statiques.

BERT est d'abord et avant tout entraîné à atteindre un objectif de *masked language modeling* (MLM). Lors de l'entraînement, une partie aléatoire des tokens d'entrée est masquée par un token spécial `<MASK>`, et BERT doit prédire le token original qui était à sa place. BERT est également entraîné à accomplir un tâche de prédiction de la phrase suivante, ou *next sentence prediction* (NSP). À l'entraînement, deux phrases séparées par un token spécial `<SEP>` sont données en entrée au modèle, et ce dernier doit choisir si la deuxième suit bel et bien la première ou non. BERT retourne un score de certitude entre 0 et 1 en plus de sa réponse. Nous faisons appel au MLM et à la tâche de NSP en totalité ou en partie pour faire générer par BERT les plus proches voisins des entrées du DS. Nous expliquons notre méthode à la section §3.2

Pour calculer les vecteurs contextuels d'une séquence linguistique, BERT commence par la segmenter en tokens. Ensuite, ces tokens sont associés à des vecteurs statiques pré-calculés, et BERT peut alors commencer à calculer les vecteurs correspondant à ces tokens en contexte, à l'aide d'un mécanisme d'attention.

En effet, pour calculer le vecteur contextuel d'un token, BERT calcule un score d'attention pour tous les autres tokens de la séquence en fonction de leur importance dans le calcul du vecteur du token en question. Lorsqu'un autre token de la phrase est utile à BERT dans le calcul adéquat du vecteur, le score d'attention est élevé, et lorsqu'un token n'est pas important pour le calcul, le score d'attention est bas.

Les scores d'attention sont ensuite combinés lors du calcul du vecteur final. Grâce à ce mécanisme, BERT peut calculer un vecteur contextualisé, en ce que le vecteur du token contient maintenant une petite partie de tous les autres tokens de la phrase dans une proportion correspondant à leur importance relative pour son calcul et l'atteinte de l'objectif. Par exemple, dans l'entraînement pour le MLM, BERT pondère le vecteur d'un token masqué selon les autres tokens de la phrase de manière à pouvoir le prédire correctement.

La figure 3.2 présente une visualisation du mécanisme d'attention. Les cases foncées correspondent à un score d'attention élevé et les cases plus pâles à un score d'attention bas.

Pour une explication plus détaillée du fonctionnement des *Transformers* et du mécanisme

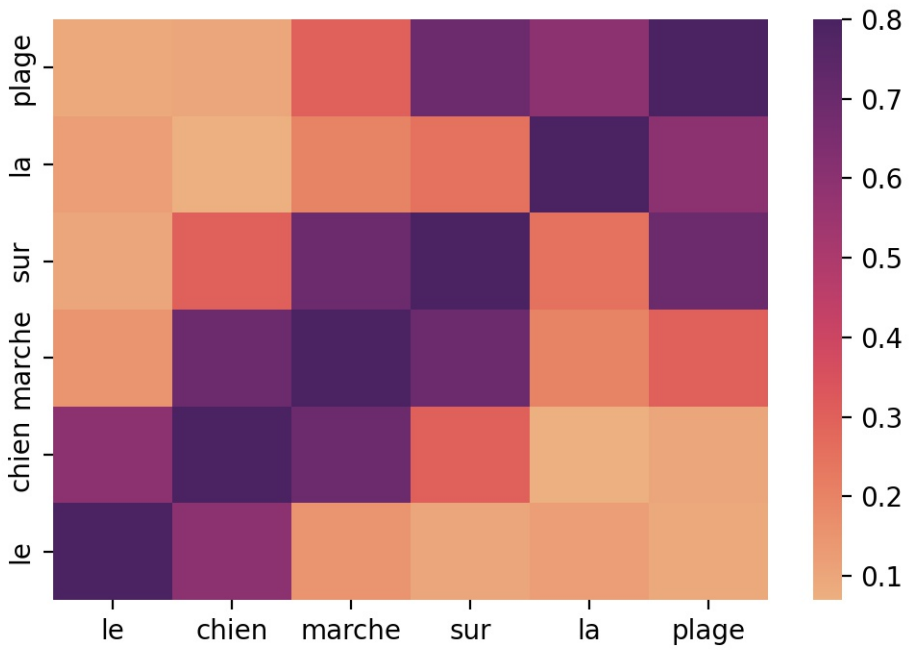


FIGURE 3.2 – Visualisation du mécanisme d'attention

d'attention, le lectorat peut se référer à Jurafsky et Martin (2009), Vaswani *et al.* (2017) ou Pilehvar et Comacho-Collados (2021).

En bref, nous avons choisi d'utiliser BERT pour générer des lexicalisations supplémentaires pour sa capacité à calculer des vecteurs en contexte plutôt que statiques. En effet, les vecteurs générés par BERT ne sont pas sémantiquement ambigus puisqu'ils sont déterminés par leur contexte, et peuvent donc être utilisés pour générer des lexicalisation supplémentaires pour les entrées du DS, qui elles aussi ne correspondent qu'à un seul sémantème chacune.

3.2 Méthodologie

La deuxième partie de la construction du DS consistait à y ajouter des lexicalisations absentes du DS mais qu'un modèle vectoriel pourrait connaître. L'objectif était donc de calculer le vecteur de chaque nœud du RL-fr dont le nom normalisé correspond à une entrée du DS et d'en récupérer les plus proches voisins, qui selon l'hypothèse distributionnelle (voir §3.1.1) devraient avoir un sens similaire et peuvent donc être considérés comme des lexicalisations exactes ou approximatives. Dans cette section, nous discutons des méthodes employées pour

y parvenir.

Dans cette partie de notre recherche, nous faisons plutôt référence aux plus proches voisins d'une lexie du DS générés par camemBERT comme des synonymes potentiels plutôt que des lexicalisations. En effet, camemBERT calcule une bonne représentation de la structure syntaxique des phrases qui lui sont données à l'entrée, ce qui fait que les tokens associés aux vecteurs calculés en contexte sont régis par la structure de la phrase. Par exemple, si on demande à camemBERT de calculer les plus proches voisins d'un nom, ceux-ci seront presque exclusivement des noms, si on lui demande de calculer les plus proches voisins d'un verbe, ceux-ci seront des verbes, etc. Les synonymes sont des mots partageant le même sens et la même partie du discours, alors que les lexicalisations sont des mots associés à un même sens, sans distinction au niveau de la partie du discours. Étant donné que les plus proches voisins d'un token calculés en contexte par camemBERT possèdent tous la même partie du discours, nous pensons qu'il est plus exact dans ce contexte de parler de synonymes plutôt que de lexicalisations, bien qu'ultimement les synonymes proposés par camemBERT pour enrichir l'ensemble des lexicalisations associé à une entrée du DS en deviendront eux-mêmes des lexicalisations.

Comme mentionné en §2.1.1, chaque nœud du RL-fr est associé à une ou plusieurs phrases-exemples. Comme les entrées du DS et les nœuds du RL-fr partagent leur nom normalisé, nous nous sommes servi des phrases-exemples pour calculer le vecteur de chaque nœud en contexte afin de trouver de nouvelles lexicalisations pour l'entrée du DS équivalente.

Un grand nombre de noms normalisés contient un code permettant de désambiguïser les différents sens d'un vocable (cf. §2.1). Nous ne pouvions donc pas directement donner le nom normalisé des entrées à camemBERT, qui ne reconnaîtrait pas cette forme. Nous devions le nettoyer et c'est le contexte contenu dans la phrase-exemple qui permettrait à camemBERT de « comprendre » le sens du token correspondant au nœud dans la phrase et ainsi d'en calculer un vecteur désambiguïsé. Voici comment nous avons procédé :

1. L'information concernant la position du nœud dans la phrase est associée aux phrases-exemples. Cette position est représentée dans le RL-fr par une paire d'entiers correspondant à la position des caractères de début et de fin de la partie de la phrase correspondant au nœud, comme présenté dans le tableau 3.2.
2. Nous avons ciblé par sa position la partie de la phrase pertinente (le nœud en contexte) et l'avons masquée.
3. Pour générer les plus proches voisins dans l'espace vectoriel du nom normalisé des

entrées du DS, nous avons fait générer par camemBERT les 10 mots dont la probabilité de se trouver à la place du masque était la plus haute.

Nœud	Phrase	Position
MANGER$_{L1a}$	Je trouvai un bar ouvert, mangeai un sandwich de pain de mie à la tomate et au thon.	(42, 49)
AGRÉABLE$_{1}$	Le barbecue prévu heureusement sous abri avait rassemblé une cinquantaine de personnes pour un souper agréable.	(118, 126)

TABLEAU 3.2 – Exemple de nœuds et leur position dans la phrase-exemple associée

Nous avons testé deux méthodes pour générer les candidats :

1. L'une consistait à simplement masquer la partie de chaque phrase-exemple correspondant au nœud. Puis, nous avons demandé à camemBERT de générer les 10 tokens les plus probables pour remplacer le masque. Nous appellerons cette méthode la méthode de base. La méthode est exemplifiée dans le listing 3.1.
2. L'autre consistait à fusionner la tâche d'entraînement pour atteindre l'objectif de MLM et la tâche de NSP (voir §3.1.3). Nous avions en effet remarqué que les candidats générés par la méthode de base s'éloignaient très rapidement du sens du token qu'ils étaient censés remplacer, ce qui en rendait une grande partie inutilisable pour enrichir le DS. Nous nous sommes donc inspirée de Qiang *et al.* (2019) et avons fait se suivre deux fois la même phrase-exemple séparée par le token <SEP> et avons masqué le nœud qui nous intéressait dans la deuxième occurrence de la phrase seulement. Cette méthode avait pour but d'évaluer si le fait d'avoir vu le nœud avant d'en générer les substituts aidait le modèle à mieux rester sur les rails du sens du nœud. Nous appellerons cette méthode la méthode <SEP>. Elle est exemplifiée dans le listing 3.2.

Listing 3.1 – Données d'entrée pour camemBERT avec la méthode de base

```
Je trouvai un bar ouvert, <MASK> un sandwich de pain de mie a la
   tomate et au thon.

Le barbecue prévu heureusement sous abri avait rassemblé une
   cinquantaine de personnes pour un souper <MASK>.
```

Les tokens proposés par camemBERT pour remplacer le masque sont toujours accompagnés d'un score de certitude entre 0 et 1.

Listing 3.2 – Données d'entrée pour camemBERT avec la méthode <SEP>

```
Je trouvai un bar ouvert, mangeai un sandwich de pain de mie a la
   tomate et au thon. <SEP> Je trouvai un bar ouvert, <MASK> un
   sandwich de pain de mie a la tomate et au thon.

Le barbecue prévu heureusement sous abri avait rassemblé une
   cinquantaine de personnes pour un souper agréable. <SEP> Le
   barbecue prévu heureusement sous abri avait rassemblé une
   cinquantaine de personnes pour un souper <MASK>.
```

Nous avons toutefois rencontré quelques problèmes lors de l'application de notre méthode.

En effet, certains nœuds du RL-fr sont des locutions et sont donc composés de plusieurs mots-formes. De plus, certains nœuds représentant de simples lexèmes s'expriment en contexte en plusieurs mots-formes, notamment les verbes fléchis à un temps composé et les verbes pronominaux. Le tableau 3.3 présente quelques exemples de ce dernier cas de figure.

Phrase	**Position**	**Séquence**
Nous avons accordé, dans les budgets votés au cours des deux dernières années, des sommes accrues, par exemple pour l'effort de construction de logements.	(5, 18)	avons accordé
Un escalier en bois mène à l'étage, les marches gémissent sous les pas d'Alice, elle se retient à la rambarde, essayant de se faire plus légère.	(85, 95)	se retient
Mais l'abréaction du médecin, pour n'être pas concomitante de celle du malade, n'en est pas moins exigée, puisqu'il faut avoir été analysé pour devenir analyste.	(121, 138)	avoir été analysé

TABLEAU 3.3 – Nœuds s'exprimant en plusieurs mots-formes en contexte

Lorsque l'on masque une partie de la phrase à camemBERT, le système considère le token <MASK> comme un seul élément, et ne peut donc proposer que des résultats composés d'un seul token également. Comme nous savions que camemBERT ne pourrait pas comprendre le sens des locutions ou des verbes conjugués à un temps composé et ne pourrait donc pas proposer de candidats adéquats, nous avons décidé de faire un tri dans les entrées DS pour lesquelles nous voulions générer des synonymes et ne garder que celles dont le nom normalisé représentait un lexème simple.

Pour éliminer les locutions, nous avons d'abord pensé à enlever les entrées dont le nom normalisé contenait une espace, puisque les locutions sont toujours composées de plus d'un lexème. Toutefois, ce faisant, nous aurions aussi perdu les entrées dont le nom normalisé correspond à un verbe pronominal.

Nous avions toujours accès à l'information interne des nœuds du RL-fr, dont leur partie du discours, et donc par correspondance à la partie du discours du nom normalisé de toutes les entrées du DS. Ainsi, pour éliminer seulement les locutions, nous avons cherché dans le RL-fr les nœuds dont la partie du discours contenait « locution », « syntagme », « phrase » ou « construction », termes indiquant tous la présence d'une locution. Ensuite, nous avons supprimé de la liste des entrées du DS pour lesquelles générer des candidats avec camemBERT celles dont le nom normalisé correspondait aux nœuds trouvés, et n'avons gardé que celles dont le nom normalisé correspondait à celui d'un lexème simple, exclusion faite des verbes pronominaux.

Pour gérer les cas des verbes pronominaux ou fléchis à un temps composé en contexte, nous avons écrit une liste exhaustive d'auxiliaires et de pronoms réfléchis du français (disponible en annexe dans la figure A.1) afin de les rechercher dans la chaîne de caractères correspondant à un nœud du RL-fr dans chaque phrase-exemple et de les ignorer au moment de la masquer. Toutefois, nous devions d'abord modifier la paire indiquant la position du nœud pour être en mesure de retrouver seulement la partie du verbe à masquer pour camemBERT, qui ne correspondrait plus à la chaîne indiquée dans le RL-fr si nous voulions ignorer les auxiliaires et les pronoms réfléchis. Pour ce faire, nous avons défini une fonction qui compte chaque caractère d'une chaîne et qui sépare les mots-forme à chaque espace. Ainsi, nous avons ciblé la partie de la phrase indiquée par la position indiquée dans le RL-fr et nous avons construit une liste contenant pour chaque mot-forme présent une paire (mot, position). Le listing 3.3 présente cette fonction et le tableau 3.4 présente le résultat obtenu pour une séquence dans une phrase-exemple en appliquant la fonction.

Nœud	ANALYSER2
Séquence	avoir été analysé
Position	(121, 138)
Résultat	[('avoir', (121, 126)), ('été', (127, 130)), ('analysé', (131, 138))]

TABLEAU 3.4 – Résultat obtenu avec l'application du script 3.3

Nous avons ensuite supprimé de la liste les paires dont le mot se retrouvait dans notre liste d'auxiliaires et de pronoms, pour ne garder que la partie lexicale d'un verbe conjugué

Listing 3.3 – Fonction find_offset

```
def find_offset(string, start):
    """
    Separates multi-word expressions at space and returns
       separated words and their individual offset - start and end
       position - in a string
    Parameters
        string (str): the string to be separated
        start (int) : starting point of the (multi) word offset,
           can be found in LN examples data
    Returns
        [(word, offset)] (list of tuples)
    """
    substr = []
    subsub = []
    unwanted = '\s+'
    for count, char in enumerate(string, start=start):
        if not re.search(unwanted, char):
            subsub.append((count, char))
        else:
            substr.append(sub)
            subsub = []
    if string[-1]:
        substr.append(sub)
    word = [[char for (off, char) in lst] for lst in substr]
    word = [''.join(char) for char in word]
    offset = [(tpl[0][0], tpl[-1][0]+1) for tpl in substr]
    word_offset = [(word[i], offset[i]) for i in range(len(word))]
    return word_offset
```

à un temps composé ou d'un verbe pronominal sans en perdre la position de début et de fin. Nous avons donc pu masquer cette partie et demander à camemBERT de nous suggérer des candidats de la même manière que nous l'avons fait avec les lexèmes ne s'exprimant qu'en un seul mot-forme.

Toutefois, nous n'avons supprimé les auxiliaires et les pronoms réfléchis que si la position du nœud dans la phrase englobait plus d'un mot-forme. En effet, nous ne voulions pas supprimer les nœuds du RL-fr correspondant à des auxiliaires, à des verbes lexicaux partageant leur forme avec les auxiliaires et à des pronoms réfléchis. Lorsque la position de ces nœuds dans la phrase n'englobe pas d'autres lexèmes, nous sommes assurée qu'il s'agit du nœud lui-même en contexte plutôt qu'un auxiliaire ou un pronom supportant un nœud.

Nous avons donc fait générer par camemBERT pour les phrases-exemples associées aux entrées que nous avions gardées les 10 candidats ayant la plus haute probabilité de se retrouver à la place du nœud dont le nom normalisé correspondait à une de ces entrées. Ensuite, nous avons procédé à leur évaluation.

3.3 Synthèse

Dans ce chapitre, nous avons présenté les plongements lexicaux, une méthode de représentation mathématique du sens basée sur l'hypothèse distributionnelle, ainsi que deux modèles vectoriels bien connus. Pour tester la possibilité d'enrichir le DS avec un modèle de langue vectoriel, nous avons choisi camemBERT, qui prend en compte le contexte du token pour lequel il calcule le vecteur. Cette capacité nous a permis de calculer des vecteurs sans ambiguïté sémantique pour les entrées du DS, qui sont des sémantèmes et sont donc, par définition, désambiguïsées.

CamemBERT est entraîné à atteindre un objectif de MLM et à accomplir une tâche de NSP. Dans l'entraînement pour le MLM, un token spécial <MASK> permet de masquer une partie du corpus et camemBERT doit prédire le token qui se trouve à la place d'un masque. Dans la tâche de NSP, un token spécial <SEP> sépare deux séquences et camemBERT doit déterminer si la seconde suit la première.

Nous avons testé deux méthodes pour générer les lexicalisations candidates avec camemBERT :

1. Dans les phrases-exemples associées aux nœuds du RL-fr, nous avons masqué le nœud dont le nom normalisé correspond à celui de chaque entrée du DS et avons demandé à camemBERT de nous proposer les 10 tokens ayant le plus de probabilités de se retrouver à la place du masque (méthode de base).
2. Nous avons d'abord montré chaque phrase-exemple complète à camemBERT, suivie du token <SEP>, puis lui avons montré la phrase à nouveau mais cette fois en masquant la partie de la phrase correspondant au nœud dont le nom standad correspond à une entrée du DS, et avons fait générer les 10 tokens ayant le plus de probabilités de se retrouver à la place du masque (méthode <SEP>).

Dans le prochain chapitre, nous présentons une évaluation en profondeur des candidats produits par camemBERT dans l'optique d'enrichir le DS.

Chapitre 4

Évaluation des lexicalisations candidates générées par camemBERT

Pour évaluer les lexicalisations candidates générées par camemBERT, nous avons d'abord voulu les comparer avec celles contenues dans le DS. En effet, comme ce dernier est basé sur des données méticuleusement compilées par des lexicographes, nous pensons que la capacité de camemBERT à recréer dans une certaine mesure le contenu d'une telle ressource donne une bonne idée de sa compréhension des sens et des relations lexicales. De plus, si nous voulons utiliser camemBERT pour nous aider à améliorer le DS, nous devons nous assurer que les candidats qu'il propose pour d'éventuelles nouvelles entrées sont adéquats. Ainsi, comparer camemBERT au DS nous permet d'évaluer sa capacité à reproduire les lexicalisations du DS qui sont, selon nous, adéquates, et donc sa capacité à produire des lexicalisations supplémentaires. Nous présentons donc en §4.1 notre méthode d'évaluation des candidats par rapport au DS et les résultats obtenus.

D'un autre côté, étant consciente que le fonctionnement de camemBERT se base peut-être sur une compréhension du sens linguistique complètement différente de celle des humains, nous avons aussi évalué les candidats générés par camemBERT qui ne se trouvaient pas dans l'ensemble de lexicalisations des entrées pour lesquelles ils ont été générés. En effet, un des objectifs de la présente recherche est de déterminer si camemBERT est capable de proposer des lexicalisations pertinentes supplémentaires pouvant enrichir celles extraites du RL-fr. Ainsi, nous devions aussi évaluer la qualité des candidats ne se trouvant pas déjà dans le DS. La section §4.2 présente la méthode d'évaluation de ces candidats et les résultats obtenus.

4.1 Comparaison avec le dictionnaire sémantique

Avant d'évaluer les candidats de camemBERT, nous devions décider avec quelles données du DS les comparer.

1. Nous avons décidé d'éliminer la lexicalisation triviale (voir p. 35) des ensembles de candidats et de l'ensemble des lexicalisations de chaque entrée du DS. En effet, que camemBERT trouve ou non la lexicalisation triviale ne nous avance pas, puisque celle-ci est toujours évidente.
2. Ensuite, nous avons évalué les candidats par rapport à un sous-ensemble du DS, basé seulement sur les FLPSV dont la valeur possédait la même partie du discours que son argument. En effet, nous avons observé que lorsque nous masquons des tokens dans une séquence, les candidats proposés par camemBERT en respectent la structure syntaxique, en ce qu'ils possèdent la majorité du temps la même partie du discours que le token masqué. Ainsi, pour que les candidats soient comparables aux lexicalisations du DS, nous avons ramené celui-ci à son niveau en limitant la partie du discours des lexicalisations à celle de la lexie dont le nom normalisé est équivalent à celui de l'entrée.
3. Ensuite, comme le nom normalisé des entrées et des lexicalisations du DS est sous une forme lemmatisée, nous avons aussi lemmatisé les candidats proposés par camemBERT pour pouvoir les comparer avec les données du DS. Nous avons utilisé pour cela la librairie FrenchLefffLemmatizer [1], elle-même basée sur le travail de Sagot (2010). Nous avons choisi ce lemmatiseur pour la grande qualité des lemmes qu'il produit, grâce entre autres à son utilisation de la partie du discours pour déterminer la forme lemmatisée. Nous avons toutefois rencontré un problème lors de la lemmatisation. En effet, en TST, le genre des noms est considéré comme un trait dérivationnel plutôt que flexionnel. Ainsi, dans le DS, le sémantème 'ouvrière' n'a pas le même sens que le sémantème 'ouvrier'. Ces deux sémantèmes sont plutôt en relation de quasi-synonymie. Toutefois, lors de l'étape de la lemmatisation, le FrenchLefffLemmatizer ramène tous les noms proposés par camemBERT à leur forme masculine. Pour que les résultats de la comparaison entre les candidats et les lexicalisations soit les plus exacts possible, nous avons extrait du RL-fr une liste des paires de noms genrés et avons empêché la lemmatisation des candidats proposés par camemBERT qui correspondaient à un nom féminin de cette liste. Nous avons ainsi pu vérifier si camemBERT est capable de trouver la forme féminine ou masculine d'un nom lorsque le contexte le requiert.

1. `https://github.com/ClaudeCoulombe/FrenchLefffLemmatizer`

4. Après avoir lemmatisé les candidats de camemBERT, nous avons nettoyé les lexicalisations contenues dans le DS. Comme nous en avons fait mention plus tôt, le nom normalisé des noeuds du RL-fr contient des chiffres et des lettres permettant de désambiguïser les différentes lexies d'un même vocable (voir le listing 4.1). Pour pouvoir comparer les candidats de camemBERT et les lexicalisations, nous devions nous assurer que si un candidat avait la même forme qu'une lexicalisation, cette correspondance puisse être détectée, ce qui serait impossible si les noms normalisés n'étaient pas nettoyés des codes permettant de désambiguïser les lexies. Toutefois, comme les lexicalisations sont contenues dans un ensemble, les lexies au nom normalisé nettoyé devenaient un seul et même vocable. Pour équilibrer du côté de camemBERT, nous avons également mis les candidats dans un ensemble pour éliminer les doublons de lemmes dûs aux différentes flexions d'un même candidat. De plus, comme nous ne pouvons savoir quel sens exprime un candidat polysémique proposé camemBERT, nous avons préféré éliminer cette distinction du côté du DS également. Ainsi, si camemBERT propose un candidat identique à une lexicalisation contenue dans le DS, nous pouvons présumer qu'il exprime une des lexies regroupées sous le vocable correspondant au nom normalisé nettoyé, et cette correspondance entre candidat et vocable montre tout de même que camemBERT peut proposer des lexicalisations pertinentes pour le DS, même si elles ne sont pas encore désambiguïsées.
5. Finalement, nous avons éliminé les ensembles de lexicalisations vides du DS pour cette partie de l'évaluation. En effet, nous considérons qu'il serait injuste d'évaluer la capacité de camemBERT à reproduire un ensemble vide alors que nous lui avons demandé de générer 10 candidats.

Listing 4.1 – Noms normalisés désambiguisant les lexèmes associés au vocable ENTRER

```
entrer#IV.2
entrer#II.2
entrer#IV.1
```

Après avoir effectué ces opérations, nous avons considéré que nous pouvions évaluer les candidats générés par camemBERT de la manière la plus réaliste possible. Les sections suivantes présentent l'évaluation de la capacité de camemBERT à reproduire le DS selon divers paramètres et perspectives.

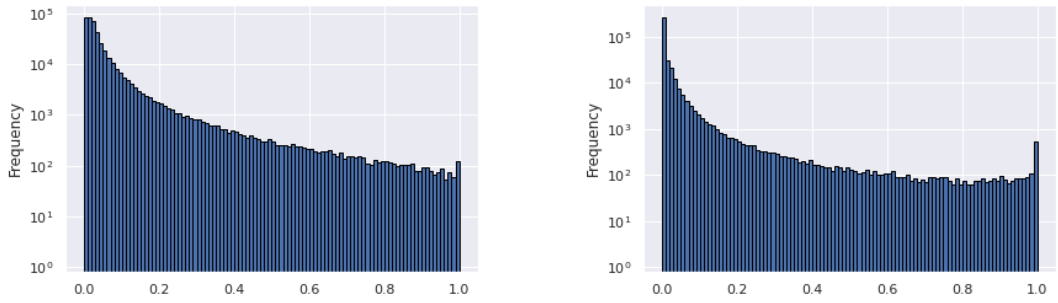
4.1.1 Le score de certitude

Nous avons commencé par observer la fréquence des scores de certitude accompagnant chacun des candidats proposés. En effet, ce score nous indique quels candidats ont la plus grande probabilité de remplacer le masque dans les phrases-exemples, et nous indique donc potentiellement lesquels sont les plus pertinents pour le DS.

Les scores de certitude sont calculés par une couche softmax, ce qui signifie que le total des scores des candidats possibles pour un masque est toujours de 1.

La figure 4.1 présente les fréquences des scores de certitude des candidats obtenus avec la méthode de base et la méthode <SEP>.

On peut y observer que, peu importe la méthode de génération des candidats employée, la grande majorité des scores sont très bas. Ainsi, avec la méthode de base, seulement 14 % des candidats ont un score de certitude supérieur à 0,1, et seulement 6 % des candidats générés avec la méthode <SEP> ont un score supérieur à 0,1. Toutefois, on observe dans la fréquence des scores obtenus avec la méthode <SEP> une certaine bi-modalité. En effet, le nombre absolu des scores les plus près de 1 obtenus avec la méthode <SEP> est plus élevé qu'avec la méthode de base. Il semble qu'avec la méthode <SEP>, le modèle est soit tout à fait sûr de son choix, soit pas du tout, mais qu'il existe moins d'entre-deux. Les scores de certitude obtenus avec la méthode de base sont, par opposition, un peu plus distribués sur tout l'axe des abscisses.



(a) Méthode de base

(b) Méthode <SEP>

FIGURE 4.1 – Fréquences des scores de certitude des candidats selon la méthode de génération

Nous avons aussi évalué la précision, le rappel et le score F des candidats proposés par camemBERT par rapport à des DS avec un PAM allant de 0 à 5 (que nous appellerons dorénavant DS 0, DS 1, DS 2, etc.). Nous avons d'abord calculé les résultats globaux de ces

Seuil	Candidats
0	[(frère, 0,1877)]
0,1	[(frère, 0,1877)]
0,3	[(frère, 0,1877), (fiancé, 0,1397)]
0,5	[(frère, 0,1877), (fiancé, 0,1397), (père, 0,0736), (ami, 0,0681), (copain, 0,0450)]
0,7	[(frère, 0,1877), (fiancé, 0,1397), (père, 0,0736), (ami, 0,0681), (copain, 0,0450), (cousin, 0,0360), (amoureux, 0,0358), (fil, 0,0333), (mari, 0,0311), (compagnon, 0,0211)]
0,9	[(frère, 0,1877), (fiancé, 0,1397), (père, 0,0736), (ami, 0,0681), (copain, 0,0450), (cousin, 0,0360), (amoureux, 0,0358), (fil, 0,0333), (mari, 0,0311), (compagnon, 0,0211)]
1,0	[(frère, 0,1877), (fiancé, 0,1397), (père, 0,0736), (ami, 0,0681), (copain, 0,0450), (cousin, 0,0360), (amoureux, 0,0358), (fil, 0,0333), (mari, 0,0311), (compagnon, 0,0211)]

TABLEAU 4.1 – Candidats pour chaque seuil de score brut pour l'entrée JUMEAU_{N}ı,ь

mesures obtenus par les candidats par rapport aux DS 0 à 5, puis en fonction de différents seuils de score brut des candidats, de leur rang et de différents seuils de leur score normalisé. En effet, comme les scores de certitude sont générés par une couche softmax, les scores de plusieurs candidats pour lesquels le modèle est aussi certain seront plus bas que celui d'un seul candidat au même niveau de certitude. Nous voulions donc rassembler dans la même catégorie de score un seul candidat dont le score est de 0,9 et trois candidats dont le score est de 0,3 chacun, puisque tous ces candidats ont en réalité le même degré de certitude. De plus, en établissant plusieurs seuils de scores et de rang, nous pouvions évaluer s'il y avait un lien entre le degré de certitude et la qualité des candidats pour le DS.

Les seuils de score brut que nous avons établis sont présentés dans le tableau 4.1. Nous avons ajouté à une liste pour chaque seuil les candidats dont la somme des scores de certitude était supérieure ou égale au seuil. Si le score d'un seul candidat dépassait le seuil de score, seul ce candidat était considéré. Si la somme des scores de certitude des 10 candidats n'atteignait pas un seuil, tous les candidats étaient ajoutés.

Ainsi, lorsque le seuil de score brut est établi à 1, tous les candidats sont considérés, et lorsqu'il est établi à 0, seul le premier choix du modèle est considéré.

Nous avons aussi comparé les candidats avec le DS en fonction de l'ordre dans lequel ils ont été produits par camemBERT (ordre décroissant du score de certitude). Nous voulions en effet inclure la notion de l'ordre des choix fait par camemBERT dans notre évaluation en plus des scores de certitude. Comme mentionné plus haut, la somme des scores de certitude des

n	Candidats
1 premier	[frère]
2 premiers	[frère, fiancé]
3 premiers	[frère, père, fiancé]
4 premiers	[frère, ami, père, fiancé]
5 premiers	[frère, père, fiancé, ami, copain]

TABLEAU 4.2 – n premiers candidats pour l'entrée JUMEAU$_{\text{N}}$**I.b** ($1 \leq n \leq 5$)

candidats générés pour un masque est toujours égale à 1. Un score de certitude bas ne signifie donc pas nécessairement que camemBERT n'est pas sûr de son choix, mais peut aussi indiquer que le modèle est certain au même degré de la validité d'un grand nombre de candidats. Nous voulions être en mesure d'identifier et d'évaluer les candidats dont camemBERT était le plus certain, même si ces derniers possédaient un score de certitude bas. Nous avons donc regroupé les candidats faisant partie des n premiers choix de camemBERT, la valeur de n allant de 1 à 5. Le tableau 4.2 présente le classement des candidats pour l'entrée JUMEAU$_{\text{N}}$**I.b**.

Nous sommes finalement arrivée à la conclusion qu'une bonne manière de combiner l'idée du score brut et du rang serait de normaliser les scores. Pour ce faire, nous avons divisé le score de chaque candidat d'un ensemble proposé par camemBERT pour une entrée du DS par le score le plus élevé de cet ensemble. Un score normalisé de 1 équivaut donc au premier choix du modèle. Nous pouvions ainsi déterminer la distance de certitude des candidats suivants par rapport au premier choix tout en gardant l'information concernant leur score. De plus, cette méthode nous permet d'estimer que les candidats avec un score normalisé élevé mais < 1, peu importe leur score brut, correspondent le plus souvent au deuxième ou troisième choix du modèle.

Nous avons établi les seuils de score normalisé présentés dans le tableau 4.3 et avons ajouté à une liste pour chaque seuil les candidats dont le score normalisé était supérieur ou égal à la valeur du seuil. Ainsi, lorsque le seuil de score normalisé est établi à 1, seul le premier choix est considéré, et lorsqu'il est établi à 0, tous les candidats sont considérés. La méthode de classement est présentée dans le tableau 4.3 pour l'entrée JUMEAU$_{\text{N}}$**I.b**.

Dans les prochaines sections, nous présentons la précision, le rappel et le score F des ensembles de candidats de camemBERT par rapport aux ensembles de lexicalisations des DS 0 à 5.

Seuil	Candidats
Seuil 0	[(frère, 1,0), (fiancé, 0,7442), (père, 0,3923), (ami, 0,3631), (copain, 0,2399), (cousin, 0,1920), (amoureux, 0,1909), (fil, 0,1776), (mari, 0,1659), (compagnon, 0,1125)]
Seuil 0,1	[(frère, 1,0), (fiancé, 0,7442), (père, 0,3923), (ami, 0,3631), (copain, 0,2399), (cousin, 0,1920), (amoureux, 0,1909), (fil, 0,1776), (mari, 0,1659), (compagnon, 0,1125)]
Seuil 0,3	[(frère, 1,0), (fiancé, 0,7442), (père, 0,3923), (ami, 0,3631)]
Seuil 0,5	[(frère, 1,0), (fiancé, 0,7442)]
Seuil 0,7	[(frère, 1,0), (fiancé, 0,7442)]
Seuil 0,9	[(frère, 1,0)]
Seuil 1,0	[(frère, 1,0)]

TABLEAU 4.3 – Candidats pour chaque seuil de score normalisé pour l'entrée JUMEAU$_{\text{N}}$**ı,ь**

4.1.2 Précision

Nous avons d'abord calculé la précision de camemBERT par rapport aux DS 0 à 5 pour les deux méthodes de génération des candidats employées.

La précision correspond à la formule :

$$\frac{Vrais \quad positifs}{Vrais \quad positifs \quad + \quad Faux \quad positifs}$$

Comme nous utilisions le DS comme le modèle à atteindre pour camemBERT, les vrais positifs correspondent aux candidats proposés pour une entrée qui se trouvent effectivement dans l'ensemble des lexicalisations de cette entrée, et la somme des vrais positifs et des faux positifs correspond en réalité à tous les candidats proposés par camemBERT, qu'ils soient dans l'ensemble des lexicalisations ou non. Nous avons donc simplifié le calcul de la précision en comptant le nombre de candidats se trouvant à la fois dans l'ensemble des lexicalisations et l'ensemble des candidats proposés pour une même entrée divisé par le nombre total des candidats, ce qui correspond à la formule suivante :

$$\frac{Intersection}{Candidats \quad proposés \quad par \quad camemBERT}$$

Pour calculer la précision globale, ou micro-moyenne de la précision du modèle par rapport au DS, nous avons additionné, pour chaque entrée, le nombre de candidats se trouvant à la fois dans l'ensemble des lexicalisations et l'ensemble des candidats proposés pour une même entrée, qui correspond à l'intersection des deux ensembles. Ensuite, nous avons fait la

somme des candidats de chaque entrée et avons divisé l'intersection par cette somme.

Entrée	Lexicalisations	Candidats	∩	Précision
PAYS$_{II}$	{royaume, lieu}	{patrie, paradis, berceau, terre, royaume, celui, capitale}	1	
CUIRASSÉ$_{ADJ}$	{blindé}	{médiéval, militaire, datant, français, gothique, défensif, royal, urbain, allemand, fort}	0	
INOUBLIABLE	{immortel, mémorable}	{acharné, mémorable, épique, terrible, intense, sanglant, singulier, difficile, final}	1	
Total	5	26	2	2/26 = 7,7 %

TABLEAU 4.4 – Extrait des données à partir desquelles la micro-moyenne de la précision a été calculée (DS 1)

Dans le tableau 4.4, la colonne « ∩ » représente le nombre de candidats présents à la fois dans l'ensemble des lexicalisations et celui des candidats pour une entrée. Nous avons fait la somme de la colonne « ∩ » puis l'avons divisée par le nombre total de candidats pour obtenir la précision. Le tableau 4.5 présente la précision obtenue avec les DS 0 à 5 selon la méthode de génération des candidats.

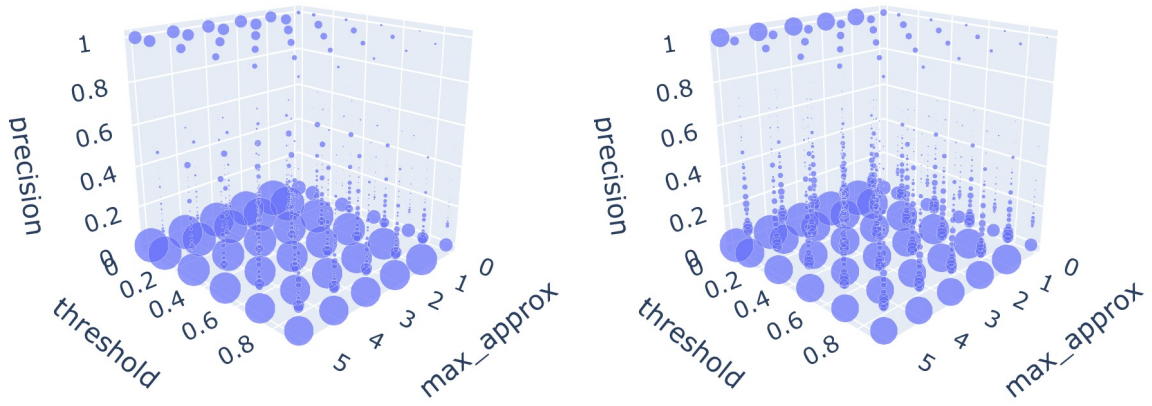
	DS 0	DS 1	DS 2	DS 3	DS 4	DS 5
Précision (méthode de base)	0,26 %	2,40 %	3,30 %	3,78 %	4,00 %	4,00 %
Précision (méthode <SEP>)	0,55 %	4,40 %	5,60 %	6,20 %	6,50 %	6,50 %

TABLEAU 4.5 – Micro-moyenne de la précision des candidats générés par camemBERT

La précision obtenue est extrêmement basse, peu importe la méthode de génération des candidats employée. Toutefois, la méthode <SEP> obtient de meilleurs résultats que la méthode de base. De plus, on remarque que plus le degré d'approximation du DS est élevé, meilleure est la précision.

Nous avons ensuite calculé la précision de chacun des candidats proposés par camemBERT individuellement en fonction de leur score brut, toujours en comparaison avec les DS 0 à 5. Nous avons mesuré la précision en établissant différents seuils de score minimaux, comme présenté dans le tableau 4.1. Chaque catégorie de seuil contient les candidats dont la

somme du score brut est égale ou dépasse la valeur du seuil. La figure 4.2 présente la précision obtenue avec les candidats proposés pour chaque entrée selon leur score brut et le DS avec lequel ils ont été comparés. Dans cette série de figures, la taille des points correspond au nombre de candidats ayant obtenu un même score de précision. Plus grand est le point, le plus nombreux sont les candidats ayant obtenu le score correspondant, toujours selon les seuils d'approximation et le DS avec lequel ils ont été comparés.



(a) Précision (méthode de base)

(b) Précision (méthode <SEP>)

FIGURE 4.2 – Distribution des scores de précision par rapport au score brut et aux DS 0 à 5

On peut faire les observations suivantes :

— Comme on peut s'y attendre suite aux données présentées dans le tableau 4.5, la précision de la majorité des ensembles de candidats générés est de 0.

— Toutefois, la précision s'élève à 1 pour une certaine partie des ensembles.

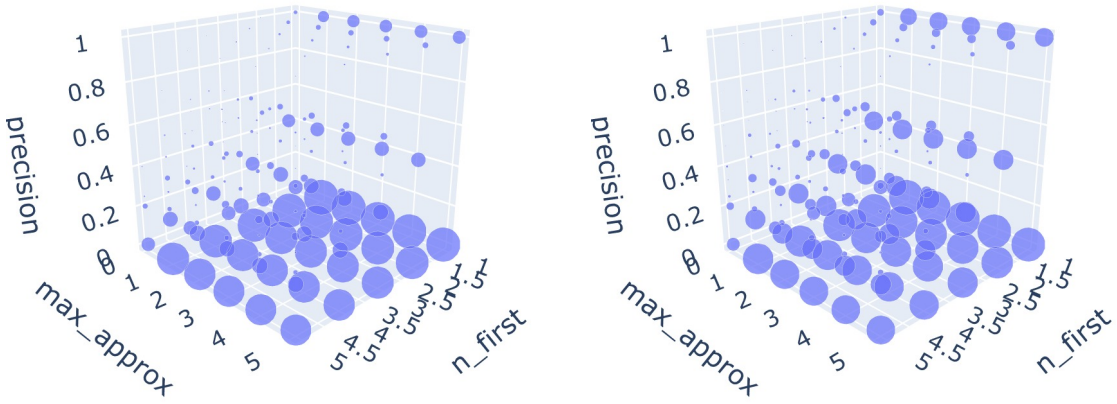
— La précision d'un plus grand nombre d'ensembles de candidats s'élève à 1 lorsque les candidats ont été générées par la méthode <SEP>.

— La précision monte plus souvent à 1 lorsque le seuil de scores bruts est de 0, c'est-à-dire lorsque seulement le premier choix du modèle est considéré, et lorsque le degré d'approximation du DS est supérieur à 0.

Nous avons ensuite calculé la précision selon le rang des candidats plutôt que leur score brut, pour inclure la notion de l'ordre des choix faits par camemBERT dans notre évaluation.

Chaque catégorie de rang contient les n-premiers choix fait par camemBERT, n allant de 1 à 5.

La figure 4.3 présente les résultats obtenus pour les candidats générés avec la méthode de base et ceux générés avec la méthode <SEP>.



(a) Précision (méthode de base)

(b) Précision (méthode <SEP>)

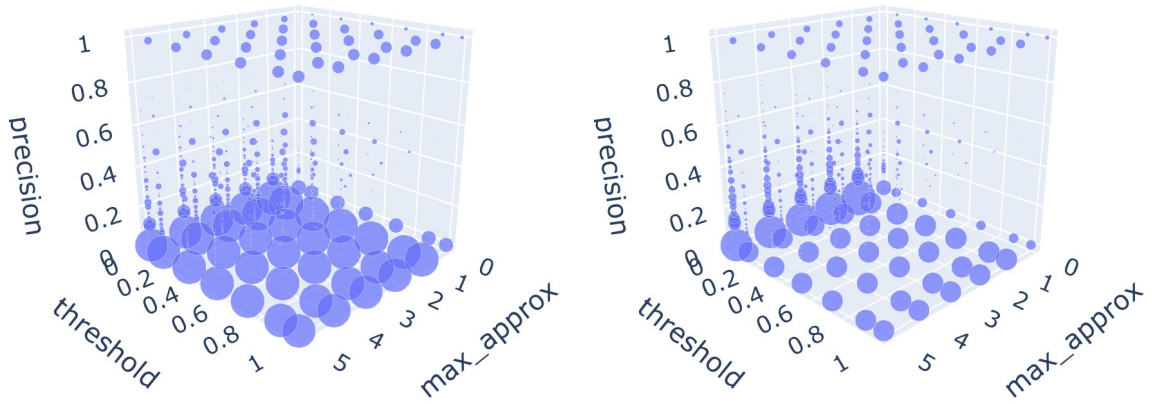
FIGURE 4.3 – Distribution des scores de précision par rapport au rang et aux DS 0 à 5

— Il apparaît encore plus clairement dans la figure 4.3 que la précision est plus souvent de 1 lorsque seulement le premier choix du modèle est considéré.

— La méthode <SEP>, encore une fois, fait augmenter la précision par rapport à la méthode de base.

— La précision est aussi plus élevée lorsque le degré d'approximation du DS est supérieur à 0.

Finalement, nous avons évalué la précision des candidats selon leur score normalisé. Chaque catégorie de seuil de score normalisé contient le ou les candidats ayant un score égal ou supérieur au seuil. La figure 4.4 présente la précision selon le seuil de score normalisé et le degré d'approximation du DS pour les deux méthodes de génération des candidats employées.



(a) Précision (méthode de base)

(b) Précision (méthode <SEP>)

FIGURE 4.4 – Distribution des scores de précision par rapport au score normalisé et aux DS 0 à 5

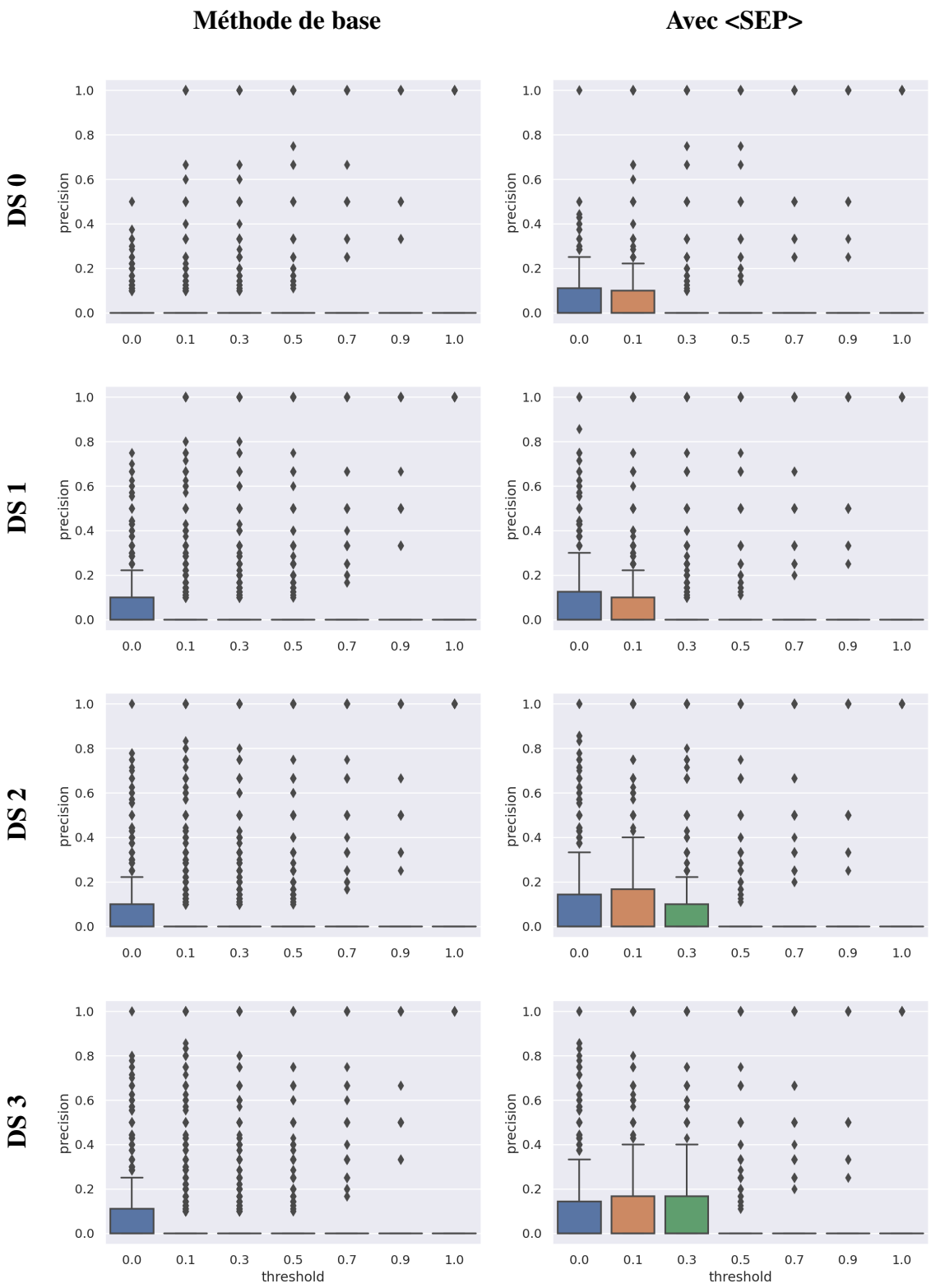
Dans la figure 4.4, on observe que :

— Le premier choix du modèle (le seul candidat ayant un score normalisé égal à 1) ne semble pas avoir un aussi gros impact sur la précision que dans les figures 4.2 et 4.3.

— Les scores de précision sont plus distribués lorsqu'on prend en compte tous les candidats (seuil = 0).

— Avec la méthode <SEP>, un moins grand nombre d'ensembles de candidats ont une précision de 0 lorsque le seuil de score normalisé > 0.

Pour mieux analyser les résultats, nous avons « découpé » les figures en « tranches » pour les observer du point de vue du seuil de score normalisé par rapport à chaque DS individuel. Nous n'avons effectué cet exercice qu'en considérant le score normalisé car nous trouvons qu'il inclut à la fois la notion de score de certitude et de rang des choix de camemBERT. La figure 4.5 présente la précision du point de vue de chaque DS individuellement par rapport aux seuils de scores normalisés. La colonne de gauche contient les résultats obtenus avec la méthode de base et la colonne de droite ceux obtenus avec la méthode <SEP>.


Méthode de base
Avec <SEP>
DS 0
DS 1
DS 2
DS 3
precision
threshold
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.1
0.3
0.5
0.7
0.9
1.0

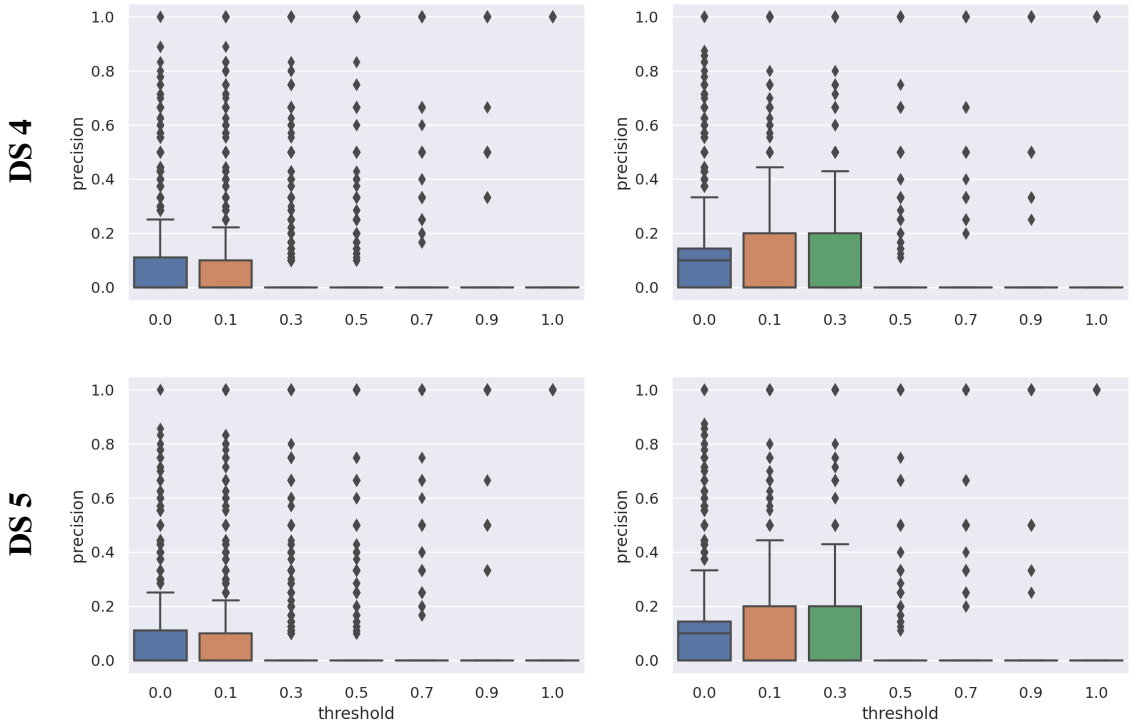


FIGURE 4.5 – Distribution des scores de précision selon les seuils de scores normalisés pour chaque DS

Les graphes présentés dans la figure 4.5 montrent les éléments suivants :

— La précision est plus élevée avec la méthode <SEP>.

— La majorité des ensembles de candidats obtient une précision supérieure à 0 lorsque le seuil de score normalisé est de 0, de 0,1 ou de 0,3, mais la précision redescend ensuite.

— La précision augmente avec le degré d'approximation du DS.

En conclusion, la précision est très basse. La méthode <SEP> génère des ensembles de candidats dont la précision est supérieure à celle des ensembles générés avec la méthode de base. Lorsqu'on considère seulement le premier choix du modèle, un plus grand nombre d'ensembles de candidats obtient une précision de 1. Cela est attendu puisque le calcul de la précision est constitué de l'intersection entre l'ensemble des candidats et celui des lexicalisations pour une entrée divisée par le nombre de candidats qui, lorsque seul le premier choix est considéré, est de 1. La précision moyenne est toutefois plus haute lorsqu'on considère presque tous les candidats (seuil de score normalisé = 0, 0,1 ou 0,3). En effet, plus grand est le nombre de candidats considérés, plus grandes sont les chances que l'un d'entre eux soit

identique à une lexicalisation du DS pour une entrée. Finalement, quand le degré d'approximation du DS est ≥ 1, la précision est améliorée. On peut poser l'hypothèse que moins les lexicalisations pour une entrée sont exactes, plus il y en a et plus grandes sont les chances pour camemBERT de trouver l'une d'elles.

La fait qu'une écrasante majorité des ensembles de candidats obtiennent une précision de 0 nous indique qu'il existe peu d'intersection entre les candidats proposés par camemBERT et les lexicalisations du RL-fr. Toutefois, nous avons conscience qu'il existe très peu de sémantèmes en français possédant 10 lexicalisations ou quasi-lexicalisations (nombre de candidats que nous avons fait générer par camemBERT pour chaque entrée), et que le calcul de la précision dresse ainsi un portrait diminué de l'intersection existant entre les deux ensembles. Heureusement, le rappel nous offre une perspective différente à cet égard.

4.1.3 Rappel

Nous avons calculé la micro-moyenne du rappel des candidats par rapport aux DS 0 à 5. Le rappel correspond à la formule suivante :

$$\frac{Vrais \quad positifs}{Vrais \quad positifs \quad + \quad Faux \quad négatifs}$$

Les vrais positifs correspondent, encore une fois, aux candidats proposés se retrouvant dans l'ensemble des lexicalisations pour une entrée, et les faux négatifs correspondent aux lexicalisations présentes dans l'ensemble que camemBERT n'a pas proposées. Ainsi, nous calculons le rappel en divisant le nombre de candidats présents à la fois dans l'ensemble de lexicalisations et l'ensemble des candidats proposés pour une même entrée par le nombre total des lexicalisations de cette entrée en utilisant la formule suivante :

$$\frac{Intersection}{Lexicalisations \quad du \quad DS}$$

Le tableau 4.6 présente la méthode que nous avons utilisée pour calculer la micro-moyenne du rappel.

Entrée	Lexicalisations	Candidats	∩	Rappel
RONFLER$_{III}$	{ronronner}	{marcher, fonctionner, manquer, tourner, sonner, tenir, rouler, caler, casser, attendre}	0	
VÉLO$_{I}$	{biclo, biclou, bicyclette, bécane, petite reine, vélocipède}	{bicyclette, pied, moto, scooter, cheval, voiture}	1	
DÎNER_N$_{II}$	{repas, souper}	{plat, repas, déjeuner, souper, diner, couscous, dessert, festin, buffet}	2	
Total	9	25	3	3/9 = 33,3 %

TABLEAU 4.6 – Extrait des données à partir desquelles la micro-moyenne du rappel a été calculée (DS 1)

Le tableau 4.7 présente la micro-moyenne du rappel obtenue pour les DS 0 à 5 et selon la méthode de génération des candidats employée.

	DS 0	DS 1	DS 2	DS 3	DS 4	DS 5
Rappel (méthode de base)	9,30 %	12,30 %	9,50 %	8,98 %	8,05 %	6,80 %
Rappel (méthode <SEP>)	17,2 %	19,6 %	14,5 %	13,00 %	11,46 %	9,9 %

TABLEAU 4.7 – Micro-moyenne du rappel des candidats générés par camemBERT

Le rappel est beaucoup plus élevé que la précision, ce qui peut s'expliquer par le fait que peu de sémantèmes possèdent 10 lexicalisations, alors que nous avons évalué la précision des ensembles de 10 candidats générés par camemBERT.

Le rappel peut être un peu amélioré lorsque nous éliminons toutes les locutions des ensembles de lexicalisations. En effet, la partie de la phrase-exemple pour laquelle camemBERT doit proposer des candidats est masquée par un seul token, <MASK>. CamemBERT ne propose donc qu'un seul token pour le remplacer, et n'a donc aucune chance de retrouver une locution présente dans l'ensemble de lexicalisations d'une entrée.

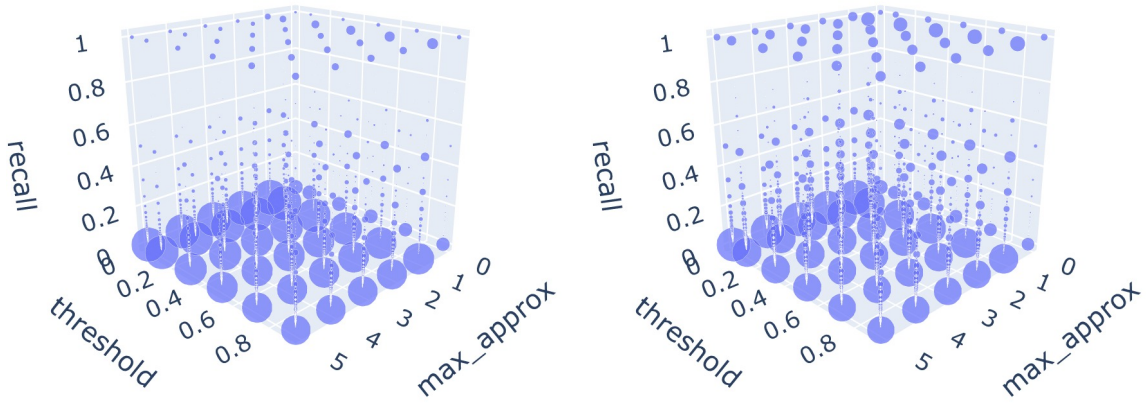
Le tableau 4.8 présente le rappel obtenu sans locutions dans le DS.

	DS 0	DS 1	DS 2	DS 3	DS 4	DS 5
Rappel (méthode de base)	10,60 %	13,90 %	10,90 %	10,33 %	9,30 %	7,90 %
Rappel (méthode <SEP>)	19,50 %	22,10 %	16,55 %	14,92 %	13,20 %	11,5 %

TABLEAU 4.8 – Micro-moyenne du rappel des candidats générés par camemBERT (sans locutions)

Toutefois, nous considérons que les locutions sont une réalité linguistique et que nous devons par conséquent évaluer camemBERT sur son incapacité à produire des lexicalisations sous forme de locutions. Le reste de l'évaluation du rappel se fera donc en gardant les locutions dans les ensembles de lexicalisations.

Ainsi, avec ou sans locutions, le rappel est maximisé lorsque les candidats sont comparés avec le DS 1, et il est plus haut encore lorsque les candidats sont générés par la méthode <SEP>. La figure 4.6 présente le rappel de chacun des candidats individuellement selon les seuils de score brut pour les deux méthodes.



(a) Méthode de base

(b) Méthode <SEP>

FIGURE 4.6 – Distribution des scores de rappel par rapport au score brut et aux DS 0 à 5

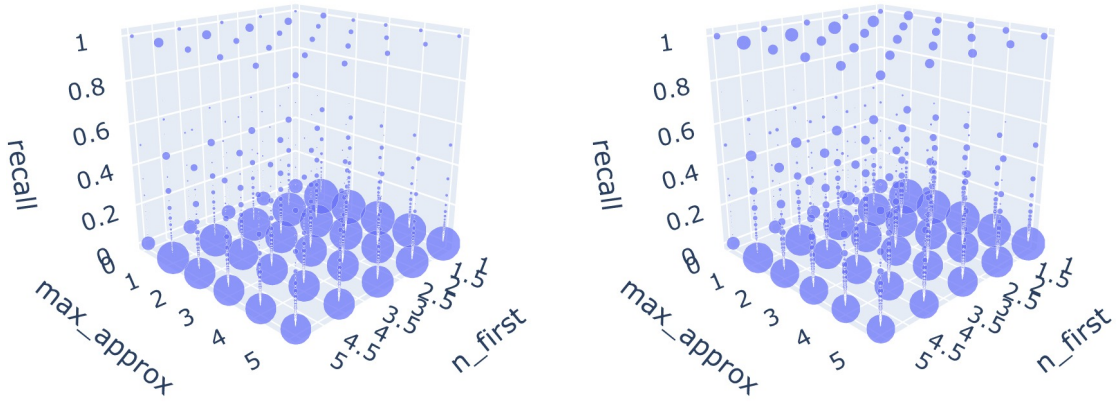
Cette figure montre les éléments suivants :

— Une grande partie des ensembles de candidats ont un rappel de 0.

— La méthode <SEP> avantage effectivement le rappel.

— Un plus grand nombre d'ensembles de candidats ont un rappel de 1 lorsqu'ils sont comparés avec les ensembles de lexicalisations du DS 1.

La figure 4.7 présente les scores de rappel en fonction du rang des candidats plutôt que des scores bruts.



(a) Méthode de base

(b) Méthode <SEP>

FIGURE 4.7 – Distribution des scores de rappel par rapport au rang et aux DS 0 à 5

En ce qui concerne le rappel, la perspective choisie (score brut ou rang) ne semble pas avoir un gros impact sur la visualisation des données. On observe les mêmes caractéristiques du rappel dans la figure 4.6 et dans la figure 4.7.

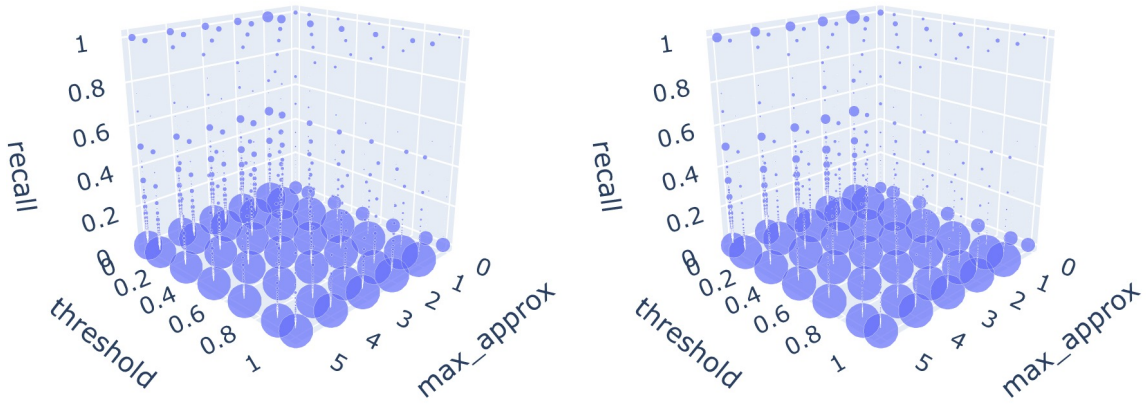
De plus, lorsqu'on ne considère que les cinq premiers choix du modèle (voir la figure 4.7) plutôt que les 10 candidats, le fait de considérer les cinq, quatre, trois ou deux premiers candidats ne semble pas faire une grande différence dans la distribution des scores de rappel. Lorsqu'on ne considère que le premier choix, par contre, le rappel diminue.

La figure 4.8 présente le rappel selon les scores normalisés des candidats.

Elle montre que :

— La majorité des ensembles de candidats obtient un rappel de 0.

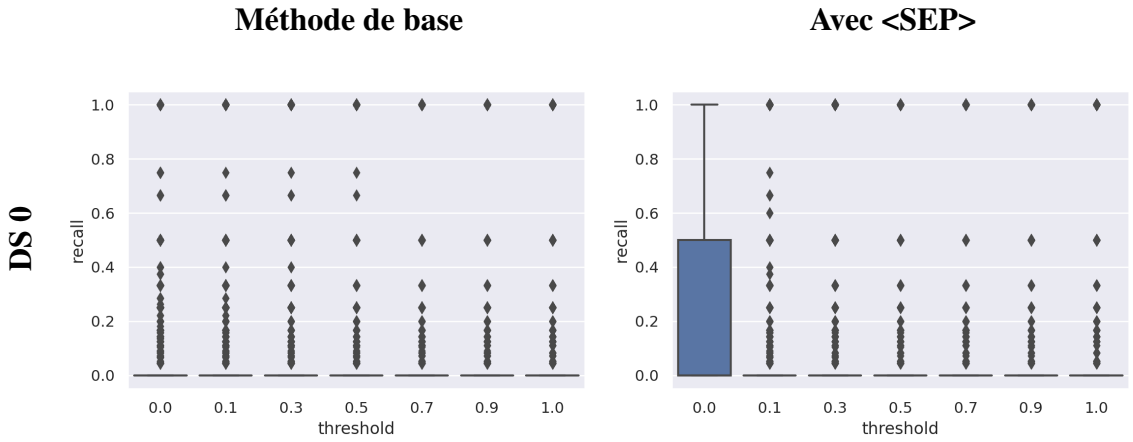
— Le nombre d'ensembles de candidats avec un rappel de 1 est le plus élevé lorsque tous les candidats sont considérés (score normalisé de 0) et lorsqu'ils sont comparés au DS 1.

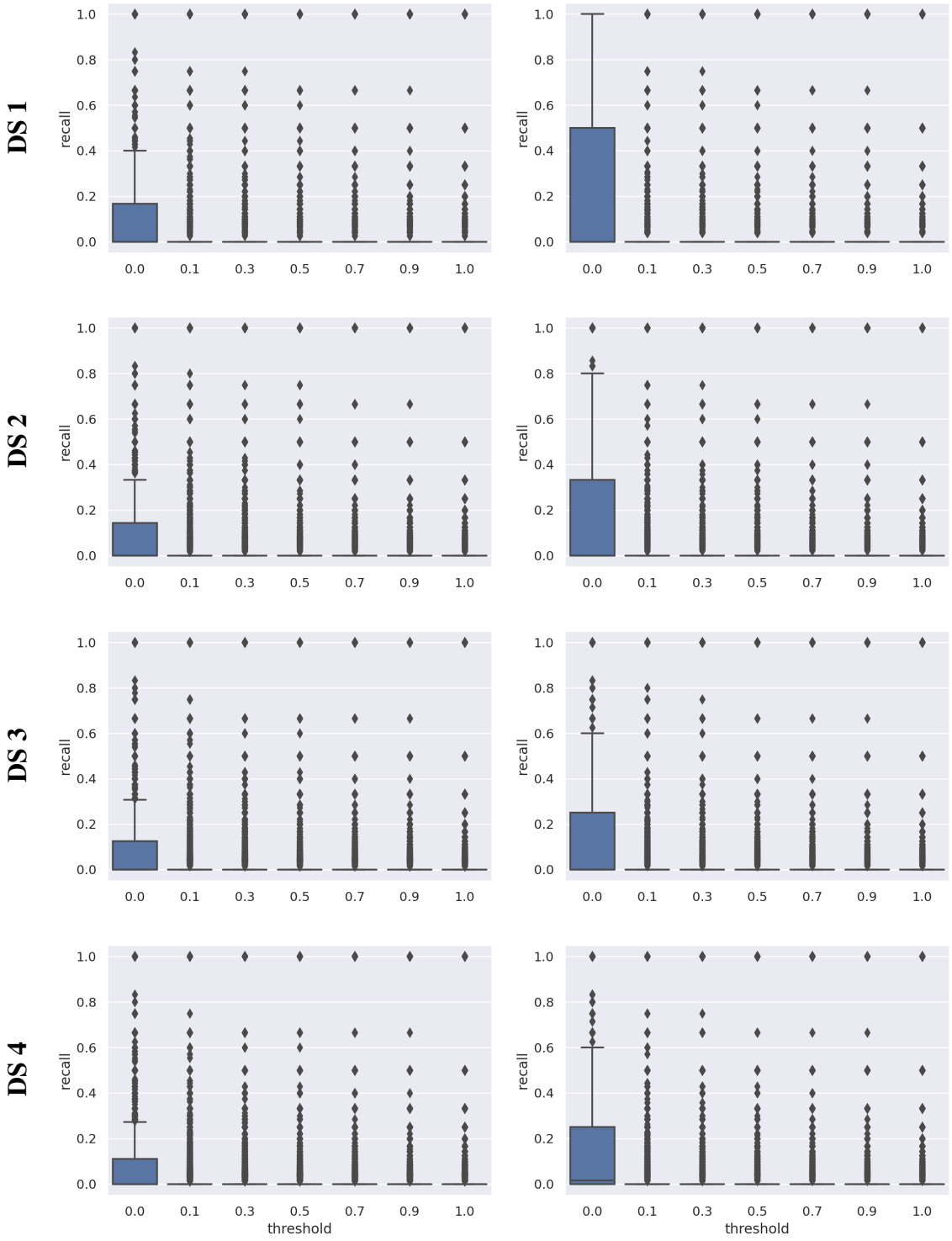


(a) Méthode de base (b) Méthode <SEP>

FIGURE 4.8 – Distribution des scores de rappel par rapport au score normalisé et aux DS 0 à 5

La figure 4.9 présente la distribution des scores de rappel selon le score normalisé du point de vue de chaque DS individuellement. À nouveau, la colonne de gauche présente les résultats obtenus avec la méthode de base et la colonne de droite ceux obtenus avec la méthode <SEP>.




DS 1
DS 2
DS 3
DS 4
recall
threshold
0.0
0.1
0.3
0.5
0.7
0.9
1.0
0.2
0.4
0.6
0.8

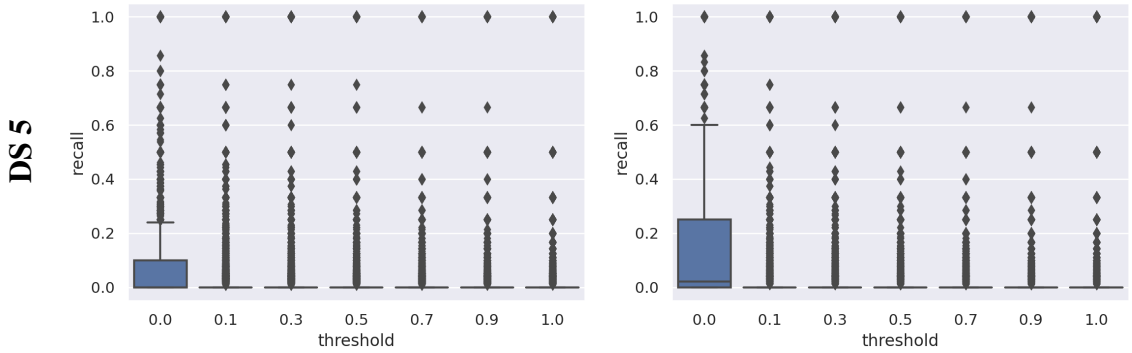


FIGURE 4.9 – Distribution des scores de rappel selon les seuils de scores normalisés pour chaque DS

On observe dans la figure 4.9 les éléments suivants :

— La méthode <SEP> avantage le rappel.

— Considérer tous les candidats (seuil = 0) avantage aussi le rappel.

— Avec la méthode <SEP>, le rappel est en moyenne le plus haut lorsque les ensembles de candidats sont comparés avec les DS 0 et 1.

— Le rappel diminue lorsque le degré d'approximation du DS augmente.

En conclusion, le rappel est aussi assez bas. Une grande partie des ensembles de candidats obtient un rappel de 0. La méthode <SEP> génère des ensembles de candidats dont le rappel est supérieur à celui obtenu par les ensembles générés avec la méthode de base. Le rappel est maximisé lorsqu'on prend en considération tous les candidats (seuil de score normalisé = 0) et que les candidats sont comparés avec les DS 0 et 1. Un plus grand nombre d'ensembles de candidats avec un rappel de 1 peut toutefois être observé lorsqu'ils sont comparés avec le DS 1. Cela peut s'expliquer par le fait que nous avons éliminé les ensembles de lexicalisations vides pour l'évaluation, qui sont plus nombreux dans le DS 0 que dans le DS 1. Nous avons donc évalué le rappel d'un moins grand nombre de données lorsque nous avons avons comparé les candidats avec le DS 0 qu'avec le DS 1.

Nous observons donc à nouveau qu'il existe peu d'intersection entre les candidats proposés par camemBERT et les ensembles de lexicalisations du DS, puisqu'un grand nombre d'ensembles de candidats obtient un rappel de 0. De plus, comme en ce qui a trait à la précision, c'est lorsque le nombre de candidats considérés est le plus élevé que le rappel moyen est maximisé. Il semble donc que la probabilité qu'un candidat soit identique à une lexicalisation pour une entrée est en partie dépendante du nombre de candidats considérés.

Nous avons aussi observé qu'un certain nombre d'ensembles de candidats obtient un rappel de 1, particulièrement lorsque tous les candidats sont considérés et qu'ils sont comparés au DS 1. CamemBERT semble donc modérément habile à recréer le DS 1.

4.1.4 Score F

Le score F est une moyenne harmonisée de la précision et du rappel, c'est-à-dire qu'il représente symétriquement à lui seul ces deux mesures. Il est calculé avec l'équation suivante :

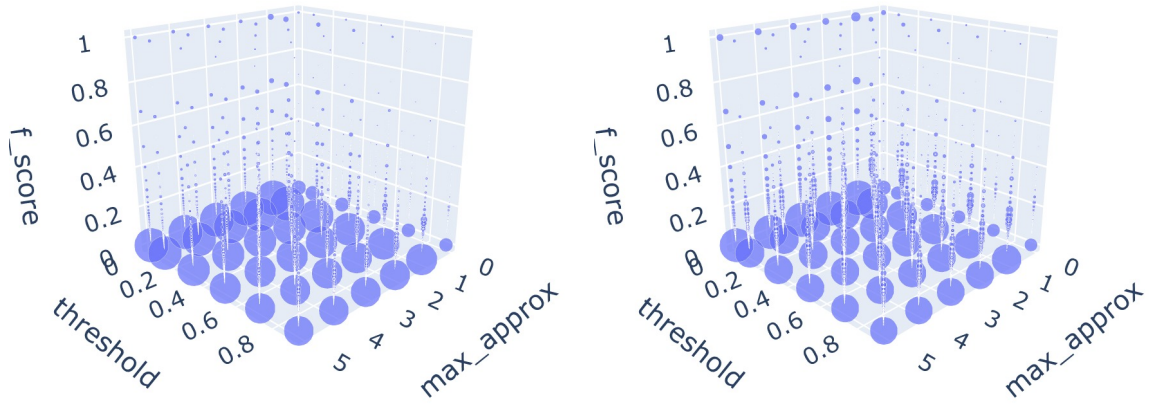
$$2 \times \frac{précision \times rappel}{précision + rappel}$$

Le tableau 4.9 présente un extrait des données utilisées pour calculer la micro-moyenne du score F.

Entrée	**Lexicalisations**	**Candidats**	∩	**Score F**
BOUCLER	{attacher}	{mettre, retirer, serrer, remettre, garder, laisser, prendre, passer, porter, nouer}	0	
CRADE1.1	{cracra, cradingue, crado, cradoque, craspec, crasseux, dégueu, dégueulasse, sale}	{glauque, dégueulasse, ridicule, moche, sale, vide, immonde, horrible, sordide, pitoyable}	2	
PANIQUE_N1	{terreur, peur}	{terreur, colère, tristesse, peur, détresse, joie, fureur, surprise, rage}	2	
Total	12	29	4	19,5 %

TABLEAU 4.9 – Extrait des données à partir desquelles la micro-moyenne du score F a été calculée (DS 1)

Nous avons calculé et visualisé le score F à partir de la précision et du rappel présentés plus haut. Comme la majorité des scores de précision et de rappel sont de 0, nous n'avons pu calculer le score F que pour une petite partie des ensembles de candidats. Tout dépendant des paramètres de score normalisé et d'approximation du DS, le plus grand nombre d'ensembles de candidats pour lesquels nous avons pu calculer le score F s'élève à 10 686, soit environ 23 % des phrases pour lesquelles les candidats ont été générés. Nous avons donc pris la décision de modifier légèrement les valeurs de la précision et du rappel. Lorsque celle-ci était



(a) Méthode de base

(b) Méthode <SEP>

FIGURE 4.10 – Distribution des scores de score F par rapport au score brut et aux DS 0 à 5

de 0, nous l'avons ajustée à 0,0001. De cette manière, le score F peut être calculé pour toutes les entrées sans qu'il n'y ait d'incidence marquée sur les résultats.

Le tableau 4.10 présente les résultats obtenus en calculant le score F de tous les candidats par rapport aux DS 0 à 5.

	DS 0	**DS 1**	**DS 2**	**DS 3**	**DS 4**	**DS 5**
Méthode de base	0,51 %	4,10 %	4,90 %	5,30 %	5,37 %	5,00 %
Méthode <SEP>	1,10 %	7,15 %	8,10 %	8,40 %	8,30 %	7,80 %

TABLEAU 4.10 – Micro-moyenne du score F des candidats générés par camemBERT

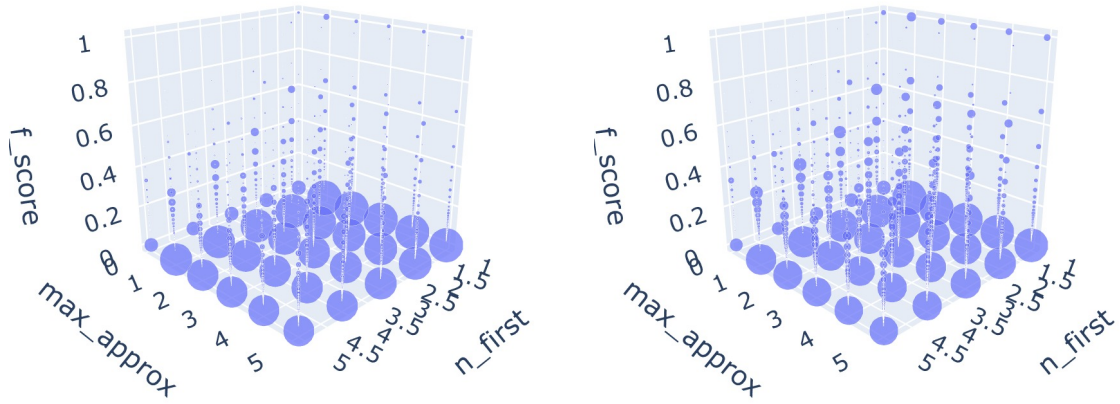
Le score F est lui aussi très bas, ce qui est attendu puisqu'il s'agit d'une moyenne de la précision et du rappel, qui sont tous les deux bas. Il atteint un plafond avec le DS 2 et reste assez stable lorsque calculé avec les DS suivants. La méthode <SEP> avantage aussi le score F.

Nous avons ensuite visualisé le score F de chacun des candidats individuellement selon les seuils de score brut et les DS 0 à 5. La figure 4.10 présente les résultats obtenus.

On y retrouve certaines caractéristiques de la précision et du rappel :

— Les résultats sont plus élevés avec la méthode <SEP>.

— Une majorité des scores F sont à 0.



(a) Méthode de base

(b) Méthode <SEP>

FIGURE 4.11 – Distribution des scores de score F par rapport au rang et aux DS 0 à 5

— Le score F atteint plus souvent 1 lorsque seul le premier choix (seuil de 0) est considéré.

— Il atteint aussi plus souvent 1 lorsqu'il est calculé par rapport au DS 1.

La figure 4.11 présente le score F selon le rang des candidats et le degré d'approximation du DS. Elle montre les éléments suivants :

— Les résultats sont toujours plus élevés avec la méthode <SEP>.

— Le score F est avantagé lorsqu'on compare les candidats avec le DS 1.

La figure 4.12 présente les résultats obtenus en calculant le score F par rapport au score normalisé des candidats et au degré d'approximation du DS.

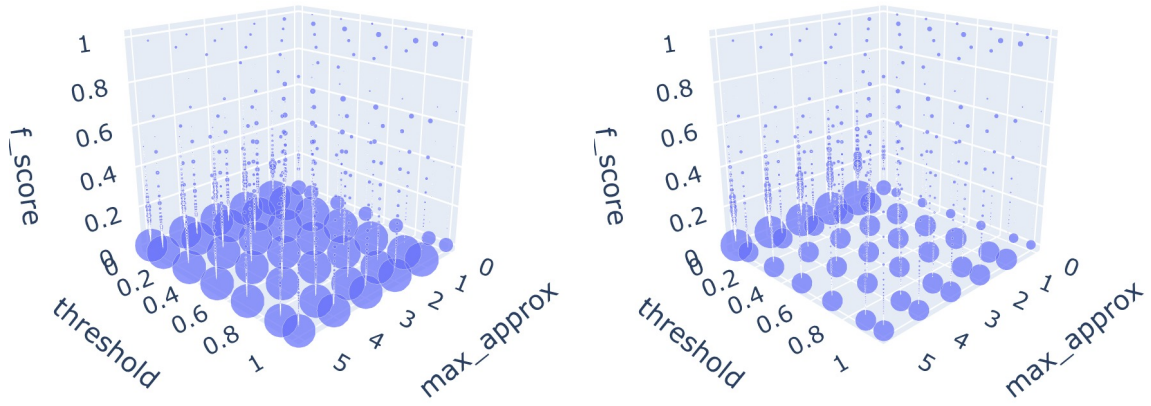
On peut y observer les éléments suivants :

— Le score F est toujours très bas, mais un nombre légèrement plus grand d'ensembles de candidats obtient un score F de 1 lorsque ce dernier est calculé par rapport au DS 1.

— Moins d'ensembles de candidats obtiennent un score F de 0 lorsqu'ils sont générés par la méthode <SEP> que par la méthode de base.

— Les scores sont plus élevés en moyenne lorsqu'on prend en compte tous les candidats (seuil = 0).

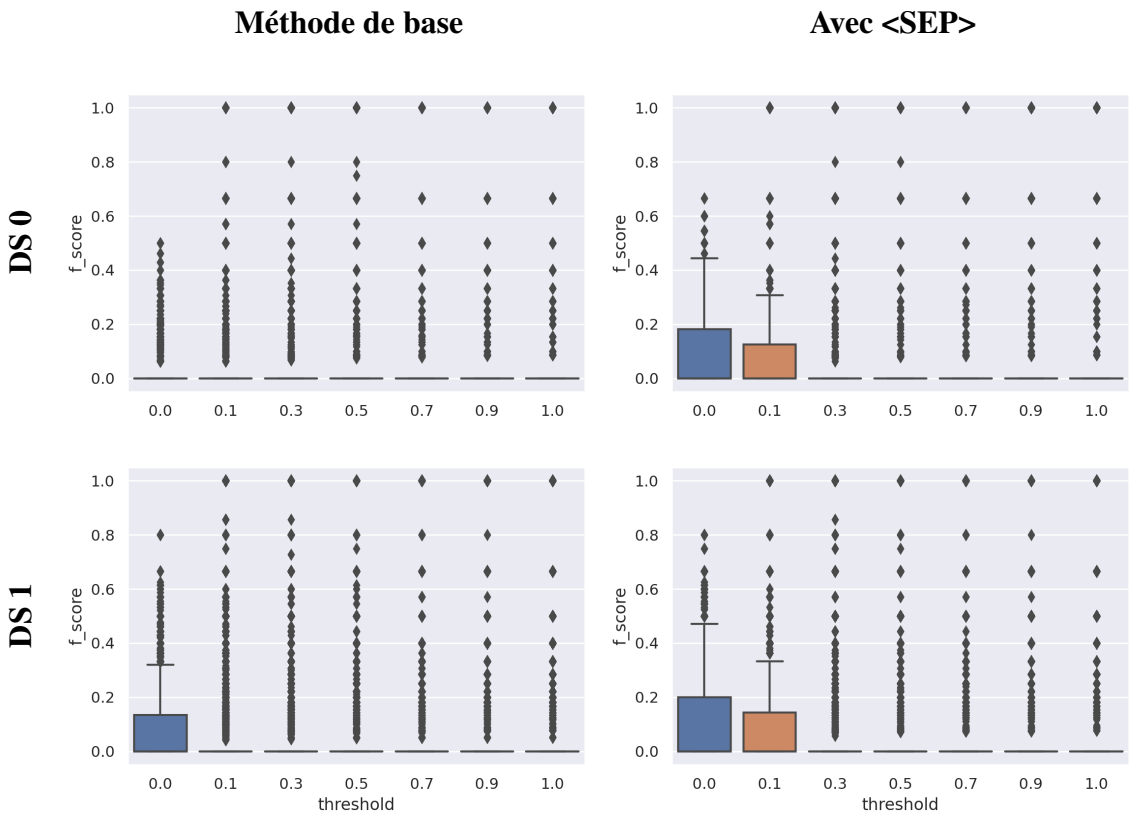
La figure 4.13 présente le score F selon chaque DS individuel, toujours par rapport aux différents seuils de score normalisé.



(a) Méthode de base

(b) Méthode <SEP>

FIGURE 4.12 – Distribution des scores de score F par rapport au score normalisé et aux DS 0 à 5



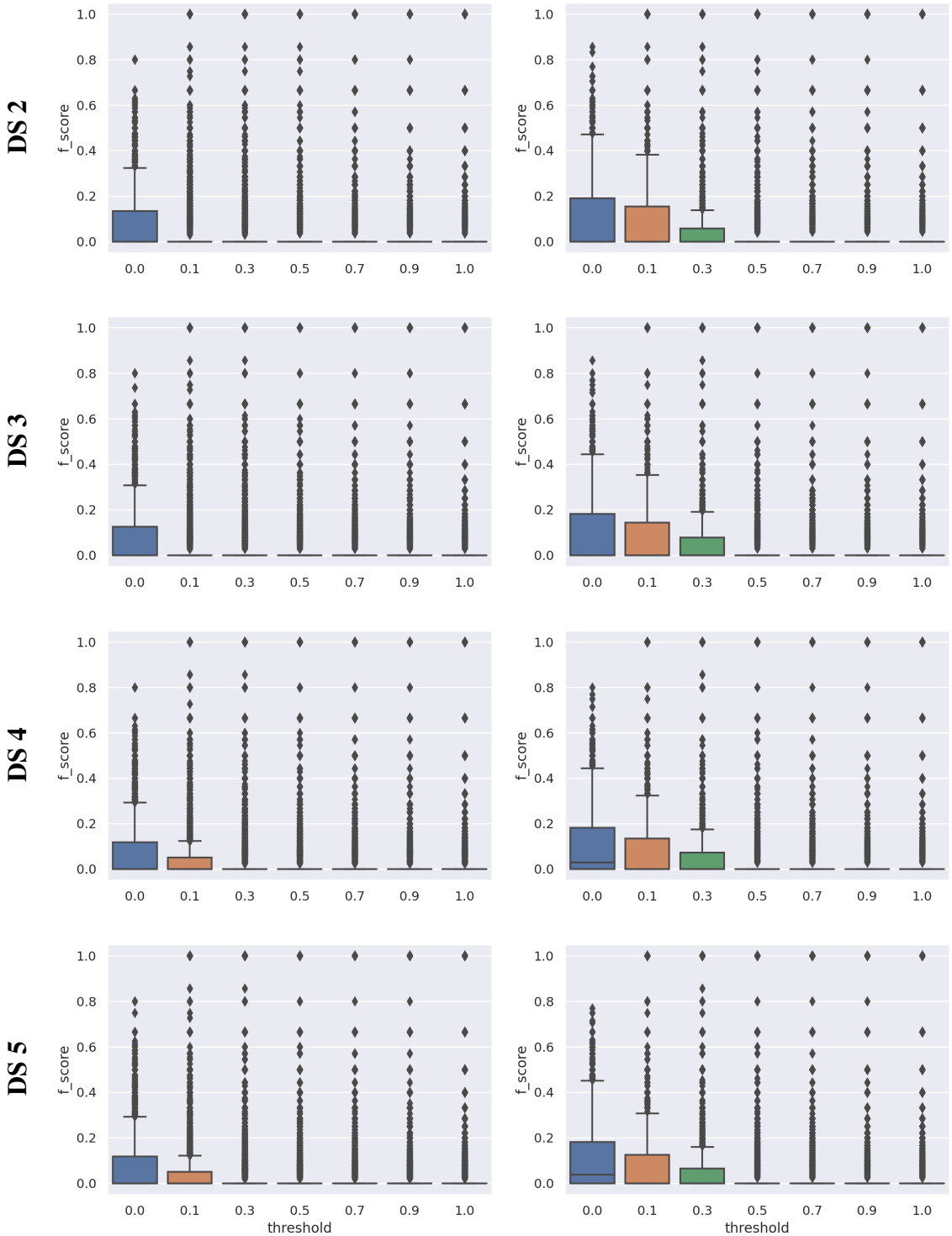


FIGURE 4.13 – Distribution des scores de score F selon les seuils de scores normalisés pour chaque DS

On y observe les éléments suivants :

— Le score F est encore très bas.

— La méthode <SEP> avantage toujours le score F.

— Le DS > 0 le fait augmenter.

— Un plus grand nombre des ensembles de candidats ont un score F supérieur à 0 lorsque le seuil de score normalisé est de 0, 0,1 et 0,3.

Le score F nous donne une idée globale de la précision et du rappel. En conclusion, le score F est très bas, bien que la méthode <SEP> permette d'obtenir des résultats un peu plus élevés. Le score F est légèrement amélioré lorsqu'il est calculé par rapport aux DS > 0 et lorsqu'on prend en considération tous les candidats (seuil de score normalisé = 0).

4.1.5 Présence dans le dictionnaire sémantique

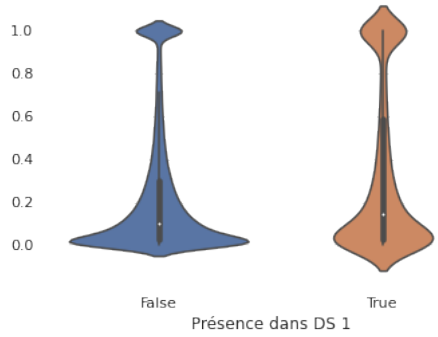
Nous avons ensuite évalué le score normalisé des candidats proposés par camemBERT selon leur présence dans le DS. Notre hypothèse était que les candidats se retrouvant dans l'ensemble des lexicalisations pour une entrée devraient avoir des scores de certitude plus hauts, et les candidats absents des scores bas. De plus, si un grand nombre des candidats ayant un score élevé et peu des candidats ayant un score bas étaient présents dans le DS, cela signifierait que les candidats au score élevé sont majoritairement des lexicalisations adéquates, et que nous pourrions nous servir de candidats ayant des scores hauts absents du DS pour l'enrichir automatiquement ou comme lexicalisations de futures entrées.

Nous avons donc évalué la présence des candidats générés avec les deux méthodes.

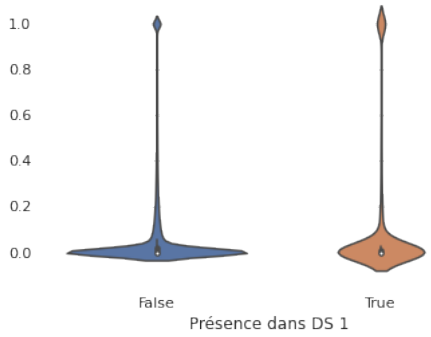
La figure 4.14 montre la distribution de tous les candidats générés par camemBERT selon leur présence dans le DS 1 et par rapport à leur score normalisé, selon la méthode de génération employée.

Nous ne présentons que les résultats obtenus avec le DS 1, car les résultats obtenus avec tous les DS sont assez semblables.

Les scores des candidats générés par la méthode de base se retrouvant dans le DS 1 sont distribués sur la majorité des valeurs possibles alors que les scores des candidats ne se trouvant pas dans le DS sont plus concentrés vers le bas. La distribution des scores normalisés des candidats générés avec la méthode <SEP>, elle, est nettement bi-modale. En effet, la plupart des valeurs se retrouvent soit à 1 soit à 0, avec très peu d'entre-deux. Cette observation fait écho à la figure 4.1, où l'on peut voir que les scores bruts des candidats générés par



(a) Présence avec la méthode de base

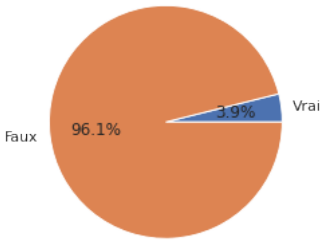


(b) Présence avec la méthode <SEP>

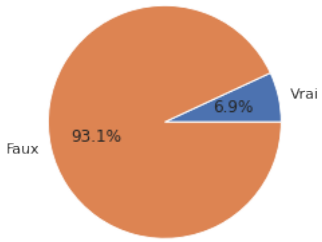
FIGURE 4.14 – Présence des candidats dans le DS 1

la méthode <SEP> sont concentrés à 0 et à 1, alors que ceux des candidats générés par la méthode de base sont plus distribués sur toutes les valeurs possibles.

Par contre, lorsqu'on observe les fréquences des candidats présents dans le DS, on s'aperçoit que bien que les candidats générés avec la méthode de base présents ont tendance à avoir des scores plus élevés, très peu du total des candidats se retrouvent en réalité dans le DS. De plus, bien que la présence des candidats générés avec la méthode <SEP> ne semble pas être très corrélée avec leur score normalisé, une plus grande proportion d'entre eux se retrouve dans le DS. La figure 4.15 nous permet de faire ce constat.



(a) Pourcentage avec la méthode de base

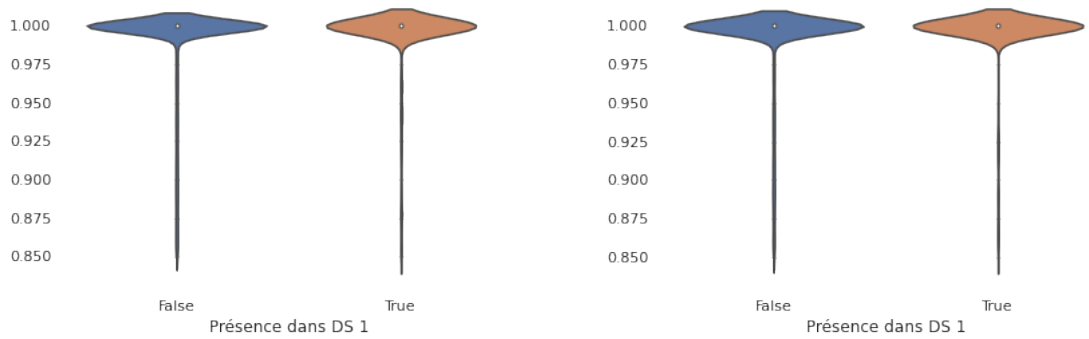


(b) Pourcentage avec la méthode <SEP>

FIGURE 4.15 – Pourcentage des candidats présents dans le DS 1

Nous avons tout de même voulu savoir si, en imposant un seuil de score normalisé minimal, le pourcentage des candidats présents dans le DS augmentait et s'il était réaliste de penser que les candidats ayant un score au-dessus d'un certain seuil étaient systématiquement

de qualité suffisante pour enrichir les lexicalisations du DS. Nous avons imposé un score arbitraire de 0,85, qui comprend tous les premiers choix du modèle (score de 1) et un grand nombre de ses deuxièmes, troisièmes et quatrièmes choix. La figure 4.16 montre la partie du haut des deux sous-figures contenues dans la figure 4.14, plus précisément la distribution des présences dans le DS 1 ayant un score de plus de 0,85.



(a) Présence avec la méthode de base (b) Présence avec la méthode <SEP>

FIGURE 4.16 – Présence des candidats dans le DS 1 avec un score normalisé ≥ 0,85

Elle montre que la distribution des candidats présents dans le DS et celle des candidats absents est très similaire pour les deux méthodes de génération des candidats. La proportion des candidats présents dans le DS est toutefois encore très différente de celle des candidats absents, comme le montre la figure 4.17.

Malgré tout, en limitant les scores à 0,85, la proportion des candidats générés avec les deux méthodes se retrouvant dans le DS augmente, en particulier celle des candidats générés avec la méthode <SEP>. En effet, le pourcentage des candidats présents dans le DS 1 passe de 6,9 % à 15,0 % lorsqu'on n'évalue que les candidats avec un score normalisé supérieur ou égal à 0,85. La proportion des candidats générés avec la méthode de base, quant à elle, passe de 3,9 % à 8,2 %. Dans les deux cas, imposer un seuil de score normalisé de 0,85 fait doubler la proportion des candidats présents dans le DS. Il semble donc bel et bien y avoir un lien entre le score normalisé et la présence des candidats dans le DS. Toutefois, même la plus grande proportion (15,0 % avec la méthode <SEP>) n'est pas suffisamment élevée pour que l'on puisse considérer que tous les candidats ayant un score normalisé ≥ 0,85 générés par la méthode <SEP> sont utilisables.

Les résultats obtenus avec les DS 0, 2, 3, 4 et 5 sont semblables à ceux obtenus avec le DS 1. Le pourcentage de présences dans un DS le plus élevé est obtenu avec une comparaison des candidats dont le score normalisé est au-dessus de 0,85 avec le DS 5. Le pourcentage

(a) Pourcentage avec la méthode de base (b) Pourcentage avec la méthode <SEP>

FIGURE 4.17 – Pourcentage des candidats présents dans le DS 1 avec un score normalisé ≥ 0,85

s'élève alors à 18,8 %, ce qui est encore trop bas pour considérer que camemBERT est capable de reproduire le DS et donc de systématiquement proposer des lexicalisations pouvant l'enrichir.

4.2 Évaluation des candidats absents du dictionnaire sémantique

Nous avons voulu explorer les raisons pour lesquelles la plupart des candidats proposés par camemBERT étaient absents du DS. En effet, sans les analyser, nous ne pouvions pas savoir s'il s'agissait de candidats pertinents pour le DS auxquels les lexicographes n'avaient pas pensé en construisant le RL-fr, ou si camemBERT ne proposait effectivement que très peu de candidats utilisables pour enrichir le DS.

Nous avons évalué manuellement un échantillon des candidats proposés par camemBERT qui ne correspondaient à aucune lexicalisation de l'entrée pour laquelle ils avaient été générés dans les DS 0 et 1 afin d'analyser leur potentiel d'enrichissement de ces DS. Nous avons choisi d'évaluer les absences du DS 0 et du DS 1 seulement car les degrés d'approximation 2, 3, 4 et 5 sont des catégories arbitraires et abstraites. En effet, nous avons considéré qu'en tant qu'humain, nous n'avons pas l'intuition linguistique nécessaire pour différencier des quasi-lexicalisations de différents degrés, mais qu'il nous est possible de faire cette distinction entre des lexicalisations exactes et des quasi-lexicalisations de degré 1.

Nous avons évalué des candidats générés par la méthode de base et la méthode <SEP>.

Pour chaque méthode, nous avons procédé de la façon suivante :

1. Nous avons divisé notre échantillon en deux catégories pour chaque dictionnaire : la moitié des candidats devait avoir un score normalisé d'au moins 0,85 sans atteindre 1 et l'autre moitié devait avoir un score normalisé de 1. Ces deux sous-catégories correspondent respectivement aux deuxième, troisième ou quatrième choix de camemBERT et à son premier choix. Nous avons choisi le seuil de 0,85 arbitrairement, en sachant qu'un seuil assez haut nous permettrait de n'évaluer que les suggestions de camemBERT dont le modèle était le plus certain. Nous voulions n'évaluer que celles-ci car la proportion des candidats présents dans le DS est plus forte lorsque leur score normalisé est supérieur ou égal à 0,85, comme démontré dans la figure 4.17. Nous pensons que ces candidats doivent donc être plus intéressants pour le DS que ceux avec un score bas, qu'il s'agisse de lexicaliser de nouvelles entrées ou d'enrichir les entrées existantes. De plus, pour des raisons de ressources et de temps, nous ne pouvions pas nous permettre d'évaluer plusieurs candidats pour un grand nombre de seuils de score normalisé dans le cadre de ce mémoire.
2. Pour chaque DS, nous avons évalué 250 candidats dont le score normalisé est supérieur ou égal à 0,85 mais inférieur à 1, et 250 candidats dont le score normalisé est exactement 1. Nous voulions évaluer un candidat absent de chaque DS pour environ 1 % des phrases-exemples associées aux entrées du DS, soit à peu près 500 candidats par DS.
3. Nous avons annoté par 0 ou par 1 si le candidat absent devrait se trouver dans l'ensemble de lexicalisations de l'entrée pour laquelle il avait été généré dans chaque DS. Ainsi, pour l'échantillon des candidats absents du DS 0, nous avons évalué si ceux-ci étaient des lexicalisations exactes de l'entrée correspondante ou non. Pour les candidats absents du DS 1, nous avons évalué s'il s'agissait de quasi-lexicalisations ou de lexicalisations exactes. Le tableau 4.11 montre un extrait des données que nous avons annotées pour le DS 0. Les données complètes se trouvent en annexe et sont présentées du tableau A.3 au tableau A.10.
4. Nous avons finalement calculé la proportion des candidats que nous jugions pertinents pour le DS correspondant. Le tableau 4.12 montre cette proportion pour la méthode de base, et le tableau 4.13 pour la méthode <SEP>.

Nous avons conscience qu'il est possible que notre seuil minimal ait été trop élevé et qu'en évaluant aussi des candidats possédant un score normalisé plus bas que 0,85, nous aurions réussi à faire grimper la proportion des candidats pertinents pour le DS. Il serait donc

Entrée	Lexicalisations	Candidat	Syn
BOUFFER[1]IV.3	MANGER_VVII	parler	0
DÉGUEULASSEVI.3	DÉGUEUVI.1	ignoble	1
S'ARRÊTERIII.2	ARRÊTERVI	cesser	1
YAOURTb	YOGOURTb	steak	0

TABLEAU 4.11 – Extrait de tableau d'évaluation des candidats absents du DS 0

intéressant, dans de futurs travaux, d'évaluer manuellement des candidats aux scores plus variés.

	DS 0	DS 1
Score normalisé ≥ 0,85 < 1	2,0 %	12,0 %
Score normalisé = 1	5,6 %	10,0 %
Total	7,6 %	22,0 %

TABLEAU 4.12 – Pourcentage des candidats absents des DS qui devraient s'y trouver (méthode de base)

	DS 0	DS 1
Score normalisé ≥ 0,85 < 1	6,5 %	14,8 %
Score normalisé = 1	9,6 %	22,0 %
Total	16,1 %	36,8 %

TABLEAU 4.13 – Pourcentage des candidats absents des DS qui devraient s'y trouver (méthode <SEP>)

Avec la méthode de base, 7,6 % des candidats dont le score normalisé ≥ 0,85 a le potentiel d'enrichir le DS 0, et 22 % a le potentiel d'enrichir le DS 1. Les candidats générés par la méthode <SEP> ont encore un meilleur potentiel d'enrichir le DS. En effet, la proportion de lexicalisations exactes susceptibles d'enrichir le DS 0 s'élève à 16,1 % et celle des quasi-lexicalisations ou lexicalisations exactes pouvant enrichir le DS 1 s'élève à 36,8 %.

Nous pouvons d'abord en conclure que, peu importe la méthode utilisée, camemBERT est plus apte à produire des quasi-synonymes du mot masqué que des synonymes exacts. Dans le cas de la méthode <SEP>, il paraît clair que le premier choix du modèle est plus

susceptible d'enrichir le DS que ses choix suivants, ce qui n'est pas aussi apparent dans le cas de la méthode de base. Cela peut s'expliquer par le fait qu'avec la méthode <SEP>, le modèle a d'abord vu le mot masqué avant de proposer des synonymes, et il est possible qu'il mise donc presque tout sur un seul candidat dont il est beaucoup plus certain que les autres. Par contre, même si la proportion des candidats pouvant enrichir le DS 1 est plus élevée que pour le DS 0, celle-ci n'est pas encore tout à fait assez élevée pour pouvoir affirmer qu'un candidat ayant un score normalisé supérieur ou égal à 0,85 devrait automatiquement être ajouté à l'ensemble des lexicalisations de l'entrée pour laquelle il a été généré. Toutefois, une ressource exploitant camemBERT pour proposer des quasi-lexicalisations aux lexicographes pendant qu'ils compilent de nouvelles entrées du DS ou en enrichissent d'anciennes pourrait être intéressante, puisque plus d'une suggestion de camemBERT sur trois générée par la méthode <SEP> avec un score normalisé supérieur ou égal à 0,85 devrait être pertinente.

4.3 Sous-segmentation des tokens

Pour comprendre pourquoi camemBERT avait autant de mal à reproduire le DS et, dans une moindre mesure, à proposer des lexicalisations pertinentes pour ce dernier, nous avons calculé la proportion des entrées pour lesquelles nous avons généré des candidats que camemBERT se représentait comme plusieurs sous-tokens. En effet, lorsque le modèle ne reconnaît pas une forme, il la segmente en plus petits tokens qu'il connaît, puis calcule le vecteur du token complet à partir de ses morceaux. Ainsi, la proportion des entrées dont le nom normalisé est sous-segmentée par camemBERT s'élève à 34,6 %. CamemBERT ne connaît donc pas 34,6 % des entrées du DS et peut donc difficilement proposer des candidats avec un sens similaire puisqu'il n'en possède pas une représentation complète.

Nous avons aussi calculé la proportion des lexicalisations des DS 0 à 5 non trouvées par camemBERT et sous-segmentées par ce dernier, excluant toujours la lexicalisation triviale. Encore une fois, si camemBERT ne possède pas de représentation pour une forme dans son vocabulaire, il lui est impossible de proposer cette dernière en remplacement du masque lors de la génération des candidats. Nous suspections que la représentation incomplète de camemBERT de ces lexicalisations était une des raisons pour lesquelles il était incapable de proposer plus de candidats présents dans le DS.

Le tableau 4.14 présente le nombre de lexicalisations que camemBERT n'a pas réussi à trouver avec la méthode de base et qu'il se représente comme plusieurs sous-tokens, ainsi que le pourcentage de ces lexicalisations qui sont des locutions et des lexèmes.

DS	Total lex. non trouvées	Sous-segmentées	Locutions	Lexèmes
DS 0	12 312	8 441	15,1 %	84,9 %
DS 1	86 151	43 431	19,7 %	80,3 %
DS 2	150 532	70 783	21,9 %	78,1 %
DS 3	188 245	86 438	22,2 %	77,8 %
DS 4	225 704	100 647	23,0 %	77,0 %
DS 5	257 649	112 667	23,5 %	76,5 %

TABLEAU 4.14 – Pourcentage des lexicalisations non trouvées par camemBERT sous-segmentées (méthode de base)

On remarque qu'environ les deux tiers des lexicalisations du DS 0 que camemBERT n'a pas trouvées sont sous-segmentées, et qu'environ la moitié des lexicalisations des DS > 0 que camemBERT n'a pas trouvées sont sous-segmentées. De plus, plus des trois quarts des lexicalisations non trouvées que camemBERT se représente comme plusieurs sous-tokens sont en fait des lexèmes, qui ne sont composés que d'un seul mot-forme. On aurait pu s'attendre à ce que camemBERT connaisse un plus grand nombre de ces formes simples.

Le tableau 4.15 présente le nombre de lexicalisations que camemBERT n'a pas trouvées avec la méthode <SEP> et qu'il se représente comme plusieurs sous-tokens, ainsi que le pourcentage de ces lexicalisations étant des locutions et des lexèmes. Le nombre de lexicalisations sous-segmentées ainsi que la proportion de ce nombre étant des lexèmes et des locutions sous-segmentés sont presque identiques aux résultats obtenus avec les candidats générés par la méthode de base. Toutefois, camemBERT réussit à retrouver plusieurs milliers de lexicalisations du DS de plus lorsque les candidats sont générés par la méthode <SEP> plutôt que par la méthode de base. Cela signifie que camemBERT réussit à trouver une plus grande proportion des lexicalisations du DS qu'il ne se représente pas comme plusieurs sous-tokens avec la méthode <SEP>. Ces données confirment que la méthode <SEP> est supérieure à la méthode de base dans la tâche demandée au modèle, et que camemBERT a beaucoup de mal à trouver les lexicalisations sous-segmentées.

DS	Total lex. non trouvées	Sous-segmentées	Locutions	Lexèmes
DS 0	11 456	8 411	15,2 %	84,8 %
DS 1	79 390	43 235	19,9 %	80,1 %
DS 2	142 138	70 553	21,9 %	78,1 %
DS 3	179 575	86 213	22,3 %	77,7 %
DS 4	216 746	100 414	23,1 %	76,9 %
DS 5	248 518	112 433	23,5 %	76,5 %

TABLEAU 4.15 – Pourcentage des lexicalisations non trouvées par camemBERT sous-segmentées (méthode <SEP>)

4.4 Synthèse

CamemBERT réussit très mal à reproduire le DS. Sa précision, son rappel et son score F par rapport à ce dernier sont en effet très bas, car les candidats qu'il propose pour une entrée se retrouvant aussi dans l'ensemble de lexicalisations associé sont peu nombreux. Les candidats que camemBERT propose pour une entrée du DS et ses lexicalisations n'ont donc presque rien en commun. On peut en conclure que camemBERT ne comprend pas vraiment le sens des entrées puisqu'il est incapable d'en générer les lexicalisations, qui sont les unités lexicales partageant leur sens avec l'entrée. CamemBERT et les modèles neuronaux semblables produisent donc bel et bien des vecteurs représentant la distribution des tokens dans un corpus plutôt que leur sens. Ainsi, on ne devrait pas les appeler « vecteurs sémantiques », puisqu'il ne représentent pas le sens linguistique.

La proportion des candidats proposés par camemBERT absents du DS et qui auraient le potentiel de l'enrichir est plus intéressante. Selon notre échantillon, avec la méthode <SEP>, qui génère les meilleurs résultats, lorsque le score normalisé du candidat est égal ou supérieur à 0,85, camemBERT propose environ une lexicalisation exacte sur six pertinente pour le DS 0, et environ une quasi-lexicalisation ou lexicalisation sur trois pertinente pour le DS 1. Cette proportion n'est toutefois pas assez élevée pour ajouter systématiquement au DS les candidats avec un score $\geq 0{,}85$ sans y introduire un grand nombre d'erreurs.

Une des raisons pour lesquelles camemBERT n'arrive pas à accomplir la tâche demandée est qu'il se représente le nom normalisé d'environ un tiers des entrées du DS comme plusieurs sous-tokens, et n'en possède donc pas une représentation complète. De plus, parmi les lexicalisations du DS que camemBERT ne trouve pas, le modèle s'en représente un grande partie comme plusieurs sous-tokens, ce qui signifie qu'il ne les connaît pas et n'aurait donc

pas pu les proposer.

Chapitre 5

Conclusion

Notre recherche visait à améliorer la capacité de lexicalisation de GenDR, un réalisateur de texte profond multilingue.

Ce réalisateur de texte, basé sur les principes de la TST, opère à l'interface sémantique-syntaxe. En effet, le système fait passer une RSém à une RSyntS, en passant par une RSyntP. Notre recherche se concentrait sur le problème de la lexicalisation profonde, c'est-à-dire, la tâche de choisir les bonnes lexies pour exprimer les sémantèmes de la RSém dans la RSyntP.

Pour effectuer la lexicalisation profonde, GenDR doit avoir accès à un DS qui établit la correspondance entre les sémantèmes d'une langue et ses lexicalisations. Plus riche est le DS, plus nombreux sont les arbres syntaxiques pouvant être générés par GenDR.

Les objectifs de cette recherche consistaient à :

1. Construire automatiquement un DS riche à partir du RL-fr afin de traiter un plus grand volume de données et d'éviter les erreurs et les incohérences qu'implique la compilation manuelle du DS.
2. Exploiter les modèles de langue neuronaux pour augmenter la couverture du DS.
3. Ajouter au module de lexicalisation de GenDR un paramètre précédemment absent permettant de régler la distance sémantique maximale entre les lexicalisations et l'entrée.

Les objectifs 1 et 3 ont été atteints. En effet, la structure du RL-fr se prête tout à fait à la construction d'un DS. Nous avons identifié un type de FL sémantiquement vides dont la valeur possède le même sens (exactement ou approximativement) que l'argument, et qui nous ont permis de lexicaliser les entrées du DS. De fait, ces FLPSV (comme nous les avons appelées) reliant certaines lexies du RL-fr entre elles nous ont permis d'extraire les nœuds possédant le même sens qu'une entrée du DS, chacune d'elle ayant été compilée à partir du nom normalisé de chacun des nœuds du RL-fr.

Nous avons également intégré au module de lexicalisation un PAM permettant d'explorer le RL-fr récursivement et d'en extraire seulement les lexicalisations dont la distance sémantique avec l'entrée est égale ou inférieure à la valeur de ce paramètre. Nous avons donc réussi à construire automatiquement un DS compatible avec GenDR contenant 29 399 entrées. Lorsque PAM = 0, le DS contient 49 235 liens de lexicalisation exacts, et lorsque PAM = 5, il contient 572 686 liens de lexicalisation et quasi-lexicalisation.

Ensuite, nous avons fait générer par le modèle de langue neuronal camemBERT des tokens candidats ayant le potentiel d'enrichir les ensembles de lexicalisations du DS. Nous avons testé deux méthodes de génération. La première (méthode de base) consistait à masquer la partie des phrases-exemples correspondant au nom normalisé des entrées en contexte et à demander à camemBERT de nous proposer les 10 tokens ayant le plus de probabilités de se trouver à la place du masque, et la seconde (méthode <SEP>) consistait à d'abord montrer la phrase complète à camemBERT avant de la masquer et de générer les candidats. Un score de certitude accompagnait tous les tokens générés.

Nous avons ensuite comparé les candidats obtenus à des DS d'approximation 0 à 5 pour les évaluer, car nous trouvions qu'une ressource basée sur les données d'une ressource lexicale compilée par des lexicographes est un bon standard auquel comparer les données générées par le modèle afin d'évaluer sa compréhension du sens lexical. Nous avons calculé la précision, le rappel et le score F des candidats par rapport aux DS et en fonction de leur score brut, de leur rang et de leur score normalisé. Les ensembles de candidats générés pour chaque entrée ont obtenu des résultats très bas pour les trois mesures d'évaluation. Nous avons ensuite mesuré la proportion des candidats se retrouvant dans le DS. Cette dernière est aussi très basse, ce qui peut expliquer les faibles scores de précision, de rappel et de score F obtenus. Toutefois, lorsque nous ne considérons que les candidats dont le score normalisé est supérieur à 0,85, cette proportion augmente considérablement. De plus, les scores de précision, de rappel et de score F obtenus et la proportion des lexicalisations trouvées par camemBERT sont toujours plus élevés lorsque les candidats ont été générés par la méthode <SEP>. Celle-ci est donc supérieure à la méthode de base, mais pas encore assez efficace pour considérer que camemBERT a une bonne compréhension du sens des entrées et des relations lexicales.

Finalement, nous avons évalué manuellement un échantillon des candidats que camemBERT avait proposés mais qui ne se trouvaient pas dans l'ensemble de lexicalisations de l'entrée associée pour déterminer si leur qualité était suffisante pour qu'ils puissent enrichir cet ensemble. Nous avons évalué les candidats dont le score normalisé était égal ou supérieur à 0,85, car l'évaluation de la présence des candidats dans le DS nous avait montré que la proportion des candidats de camemBERT se retrouvant dans le DS augmentait lorsque

leur score normalisé était égal ou supérieur à 0,85. Nous en avons donc déduit que ces candidats étaient de meilleure qualité et qu'avec des ressources limitées, il valait mieux s'en tenir à l'évaluation de ceux-ci plutôt que de tous les candidats. Les résultats obtenus indiquent qu'en générant les candidats avec la méthode <SEP>, qui obtient de meilleurs résultats, environ un candidat sur six proposé par camemBERT peut enrichir le DS 0, qui ne contient que des lexicalisations exactes, et environ un candidat sur trois peut enrichir le DS 1 contenant aussi des quasi-lexicalisations. Ces proportions, bien que plus encourageantes que les résultats obtenus pour la précision, le rappel et le score F, ne sont pas tout à fait assez élevées pour ajouter systématiquement au DS les candidats dont le score normalisé est supérieur ou égal à 0,85 sans y introduire d'erreurs. Nous imaginons toutefois qu'un outil exploitant les données produites par camemBERT et proposant des lexicalisations candidates à des lexicographes créant manuellement de nouvelles entrées dans le DS ou en enrichissant des anciennes pourrait être d'une certaine utilité. Nous pouvons en conclure que l'objectif 2 n'a été que partiellement atteint.

Pour terminer, nous croyons qu'il pourrait être pertinent d'affiner camemBERT sur les données du RL-fr avant de lui faire générer les candidats. En effet, nous pensons que les résultats de l'évaluation pourraient être améliorés si camemBERT avait d'abord accès aux données à partir desquelles nous lui faisons générer les candidats.

Il serait également intéressant de reproduire cette recherche avec des modèles de langue neuronaux plus récents et entraînés sur des corpus plus étendus. En effet, au moment d'entreprendre ces travaux, camemBERT faisait encore partie des modèles à la pointe du domaine pour le français. Au moment d'écrire ces lignes, par contre, plusieurs nouveaux modèles ont fait leur apparition, et il est tout à fait réaliste de penser qu'ils obtiendraient de meilleurs résultats que camemBERT pour la tâche que nous lui avons demandée.

Bibliographie

ABDI, H. (2007). Singular Value Decomposition (SVD) and Generalized Singular Value Decomposition (GSVD).

APRESJAN, J. (2000). *Systematic lexicography*. Oxford linguistics. Oxford University Press, Oxford ; New York.

BOHNET, B., LANGJAHR, A. et WANNER, L. (2000). A development environment for an MTT-based sentence generator. Dans *Proceedings of the first international conference on Natural language generation - INLG '00*, vol. 14, p. 260, Mitzpe Ramon, Israel. Association for Computational Linguistics.

BOHNET, B. et WANNER, L. (2010). Open soucre graph transducer interpreter and grammar development environment. Dans CALZOLARI, N., CHOUKRI, K., MAEGAARD, B., MARIANI, J., ODIJK, J., PIPERIDIS, S., ROSNER, M. et TAPIAS, D., dir., *Proceedings of the International Conference on Language Resources and Evaluation (LREC) 2010*, pp. 17–23, Valletta, Malte. European Language Resources Association.

CARROLL, J. D. et CHANG, J.-J. (1970). Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319.

de SAUSSURE, F. et de MAURO, T. (1994). *Cours de linguistique générale*. Bibliothèque scientifique Payot. Payot, Paris, Critique éd.

DEVLIN, J., CHANG, M.-W., LEE, K. et TOUTANOVA, K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. Dans *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

DUBINSKAITĖ, I. (2017). Développement de ressources lituaniennes pour un générateur automatique de texte multilingue. Mémoire de maîtrise, Université de Montréal.

DUBÉ, M. (2021). Le traitement des locutions en génération automatique de texte multilingue. Mémoire de maîtrise, Université de Montréal.

FELLBAUM, C. (1998). *WordNet*. MIT Press, Cambridge, Mass, 1.6 éd. OCLC : 40066017.

FIRTH, J. R. (1957). A Synopsys of Linguistic Theory 1930-1955. Dans *Studies in Linguistic Analysis*, vol. 1. Philological Society, Oxford. Réimprimé dans Palmer, F. (éd. 1968) Selected Papers of J. R. Firth, Longman, Harlow.

GALARRETA-PIQUETTE, D. (2018). Intégration de VerbNet dans un réalisateur profond. Mémoire de maîtrise, Université de Montréal.

HARRIS, Z. S. (1954). Distributional structure. *Word*, 10(2–3):146–162.

HOTELLING, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441.

JURAFSKY, D. et MARTIN, J. H. (2009). *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall series in artificial intelligence. Pearson Prentice Hall, Upper Saddle River, N.J, 2e éd. OCLC : 213375806.

KAHANE, S. (2003). The Meaning-Text Theory. *Dependency and Valency. An International Handbook of Contemporary Research*, 1:546–570.

LAMBREY, F. (2016). Implémentation des collocations pour la réalisation de texte multilingue. Mémoire de maîtrise, Université de Montréal.

LAMBREY, F. et LAREAU, F. (2015). Le traitement des collocations en génération de texte multilingue. Dans *Actes de la 22^{e} conférence sur le Traitement Automatique des Langues Naturelles. Articles courts*, pp. 263–269, Caen, France. ATALA.

LAREAU, F., LAMBREY, F., DUBINSKAITĖ, I., GALARRETA-PIQUETTE, D. et NEJAT, M. (2018). GenDR : A generic deep realizer with complex lexicalization. Dans *Proceedings of 11th edition of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japon.

MARTIN, L., MULLER, B., ORTIZ SUÁREZ, P. J., DUPONT, Y., ROMARY, L., de la CLERGERIE, E., SEDDAH, D. et SAGOT, B. (2020). CamemBERT : a Tasty French Language Model. Dans *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7203–7219, Online. Association for Computational Linguistics.

MEL'ČUK, I. (1981). Meaning-Text Models : A recent trend in Soviet linguistics. *Annual Review of Anthropology*, 10(1):27–62.

MEL'ČUK, I. (1995). The future of the lexicon in linguistic description and the explanatory combinatorial dictionary. *Linguistics in the morning calm*, 3:181–270.

MEL'ČUK, I. (2004). Actants in semantics and syntax I : actants in semantics. *Linguistics*, 42(1):1–66.

MEL'ČUK, I. (2012). *Semantics : from meaning to text*. Num. 129, 135, 168 de Studies in language companion series. John Benjamins, Amsterdam ; Philadelphia.

MEL'ČUK, I. (2016). *Language : from meaning to text*. LRC Publishing House ; Academic Studies Press, Moscow : Boston, Beck, David éd. OCLC : ocn932263842.

MEL'ČUK, I. et POLGUÈRE, A. (2008). Prédicats et quasi-prédicats sémantiques dans une perspective lexicographique. *Lidil*, 37:99–114.

MEL'ČUK, I. et POLGUÈRE, A. (2021). Les fonctions lexicales dernier cri. Dans MARENGO, S., dir., *La Théorie Sens-Texte. Concepts-clés et applications*, Dixit Grammatica, pp. 75–155. L'Harmattan.

MEL'ČUK, I., POLGUÈRE, A. et CLAS, A. (1995). *Introduction à la lexicologie explicative et combinatoire*. Duculot, Louvain-la-Neuve.

MIKOLOV, T., CHEN, K., CORRADO, G. et DEAN, J. (2013). Efficient Estimation of Word Representations in Vector Space. *arXiv*.

MILIĆEVIĆ, J. (2006). A Short Guide to the Meaning-Text Theory. *Journal of Koralex*, 8:187–233.

MILLER, G. A. (1995). WordNet : A Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

PEARSON, K. (1901). LIII. *On lines and planes of closest fit to systems of points in space*. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11): 559–572.

PEREC, G. (1978). *La vie mode d'emploi*. Le livre de poche. P.O.L.-Hachette, Paris.

PIERRAT, E. (2008). *Troublé de l'éveil*. Fayard, Paris.

PILEHVAR, M. T. et COMACHO-COLLADOS, J. (2021). *Embeddings in natural language processing : theory and advances in vector representations of meaning*. Num. 47 de Synthesis lectures on human language technologies. Morgan & Claypool Publishers, San Rafael, California.

POLGUÈRE, A. (1990). *Structuration et mise en jeu procédurale d'un modèle linguistique déclaratif dans un cadre de génération de texte*. Thèse de doctorat, Université de Montréal.

POLGUÈRE, A. (1998a). Pour un modèle stratifié de la lexicalisation en génération de texte. *TAL*, 39(2):57–76.

POLGUÈRE, A. (1998b). La théorie Sens-Texte. *Dialangue*, 8-9:9–30.

POLGUÈRE, A. (2000). A "natural" lexicalization model for language generation. Dans *Proceedings of the Fourth Symposium on Natural Language Processing*, pp. 37–50, Chiang-mai, Thaïlande.

POLGUÈRE, A. (2009). Lexical systems : graph models of natural language lexicons. *Language Resources and Evaluation*, 43(1):41–55.

POLGUÈRE, A. (2014). From writing dictionaries to weaving lexical networks. *International Journal of Lexicography*, 27(4):396–418.

QIANG, J., LI, Y., ZHU, Y., YUAN, Y. et WU, X. (2019). Lexical simplification with pretrained encoders. Dans *AAAI Conference on Artificial Intelligence*.

SAGOT, B. (2010). The Lefff, a Freely Available and Large-coverage Morphological and Syntactic Lexicon for French. Dans *International Conference on Language Resources and Evaluation*.

STEWART, G. W. (1993). On the Early History of the Singular Value Decomposition. *SIAM Review*, 35(4):551–566.

TESNIÈRE, L. (1959). *Élements de syntaxe structurale*. Paris, Klincksieck éd.

van der MAATEN, L. et HINTON, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605.

VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L. et POLOSUKHIN, I. (2017). Attention is all you need. Dans *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, p. 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

WANNER, L. (1996). Lexical choice in text generation and machine translation. *Machine Translation*, 11(1-3):3–35.

WANNER, L., NICKLASS, D., PANIGHI, A., PARISIO, S., SCHEU-HACHTEL, H., SERPA, J., BOUAYAD-AGHA, N., BOHNET, B., BRONDER, J., FERREIRA, F. R., FRIEDRICH, R., KARPPINEN, A., LAREAU, F. et LOHMEYER, A. (2007). From measurement data to environmental information : MARQUIS - a Multimodal AiR QUality Information Service for the general public. Dans *In Proceedings of ISESS 2007*.

WATTS, D. J. et STROGATZ, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442.

ZHAO, X. (2018). Les collocations du champ sémantique des émotions en mandarin. Mémoire de maîtrise, Université de Montréal.

ŽOLKOVSKIJ, A. et MEL'ČUK, I. (1967). O sisteme semantičeskogo sinteza. II : Pravila preobrazovanij. *Naučno-texničeskaja informacija*, 2:17–27.

Annexe A

Données

Ex. rég.	Approx.	Même PDD
^A_\d$	0	0
^A_\dPred$	0	0
^Adv_\d$	0	0
^Adv_\d v A_\d$	0	0
^Conv_\d+$	0	1
^Conv_\d+Pred$	0	0
^Pred$	0	0
^S_0$	0	0
^S_0A_\d$	0	0
^S_0Conv_\d+$	0	0
^S_0Pred$	0	0
^S_\d v S_0$	0	0
^Syn$	0	1
^V_0$	0	0
^V_0Conv_\d+$	0	0
^Figur$	0	0
^A_\d[∩⊂⊃]$	1	0
^A_\d\d[∩⊂⊃]$	1	0
^Adv_0[∩⊂⊃]$	1	0
^Conv_d+[∩⊂⊃]$	1	1
^Gener$	1	0

^S_0\^usual$	1	0
^S_0[∩⊂⊃]$	1	0
^S_0Pred_[∩⊂⊃]$	1	0
^Syn_[∩⊂⊃]$	1	1
^V_0[∩⊂⊃]$	1	0
^Syn_[∩⊂⊃]\^.+$	1	1

TABLEAU A.1 – Patrons des FLPSV compilés

Nom	Type	Approx.	Même PDD
Syn	paradigmatic	0	1
Syn_⊂	paradigmatic	1	1
Syn_⊃	paradigmatic	1	1
Syn_∩	paradigmatic	1	1
Syn_⊃^sex	paradigmatic	1	1
Syn_⊂^sex	paradigmatic	1	1
Conv_2	paradigmatic	0	1
Conv_21	paradigmatic	0	1
Conv_213	paradigmatic	0	1
Conv_23	paradigmatic	0	1
Conv_231	paradigmatic	0	1
Conv_312	paradigmatic	0	1
Conv_32	paradigmatic	0	1
Conv_321	paradigmatic	0	1
Conv_3214	paradigmatic	0	1
Conv_423	paradigmatic	0	1
Gener	paradigmatic	1	0
Figur	paradigmatic	0	0
S_0	paradigmatic	0	0
S_0_⊂	paradigmatic	1	0
S_0_⊃	paradigmatic	1	0
S_0_∩	paradigmatic	1	0
S_1 ∨ S_0	paradigmatic	0	0
S_2 ∨ S_0	paradigmatic	0	0

S_3 ∨ S_0	paradigmatic	0	0
S_4 ∨ S_0	paradigmatic	0	0
S_0Conv_21	paradigmatic	0	0
S_0Pred	paradigmatic	0	0
S_0Pred_⊂	paradigmatic	1	0
S_0Pred_∩	paradigmatic	1	0
S_0 ^ usual	paradigmatic	1	0
S_0Conv_213	paradigmatic	0	0
V_0	paradigmatic	0	0
V_0_⊃	paradigmatic	1	0
V_0_⊂	paradigmatic	1	0
V_0_∩	paradigmatic	1	0
V_0Conv_21	paradigmatic	0	0
V_0Conv_213	paradigmatic	0	0
V_0Conv_413	paradigmatic	0	0
A_0	paradigmatic	0	0
A_0_⊃	paradigmatic	1	0
A_0_⊂	paradigmatic	1	0
A_0_∩	paradigmatic	1	0
Adv_0 v A_0	paradigmatic	0	0
Adv_0	paradigmatic	0	0
Adv_0_∩	paradigmatic	1	0
Adv_0_⊃	paradigmatic	1	0
A_1	paradigmatic	0	0
A_1_⊂	paradigmatic	1	0
A_1_⊃	paradigmatic	1	0
A_1_∩	paradigmatic	1	0
A_1/2_⊂	paradigmatic	1	0
Adv_1 v A_1	paradigmatic	0	0
A_1Pred	paradigmatic	0	0
S_0A_1	paradigmatic	0	0
A_2	paradigmatic	0	0
A_2_∩	paradigmatic	1	0
Adv_2 v A_2	paradigmatic	0	0

A_3	paradigmatic	0	0
A_4	paradigmatic	0	0
Adv_1	syntagmatic	0	0
Adv_2	syntagmatic	0	0
Pred	syntagmatic	0	0
Conv_21Pred	syntagmatic	0	0

TABLEAU A.2 – FLPSV extraites du RL-fr à l'aide des patrons

avoir	s’aura	s’est	fut
ai	aurons	sommes	furent
m’ai	aurez	êtes	eu
as	auront	sont	eut
t’as	s’auront	étais	eût
a	aurais	m’étais	s’eut
s’a	m’aurais	t’étais	s’eût
avons	t’aurais	était	eurent
avez	aurait	s’était	s’eurent
ont	s’aurait	étions	été
s’ont	aurions	étiez	ayant
avais	auriez	étaient	étant
m’avais	auraient	s’étaient	s’ayant
t’avais	s’auraient	serai	s’étant
avait	aie	seras	vais
s’avait	ait	sera	vas
avions	aies	serons	va
aviez	ayons	serez	allons
avaient	ayez	seront	allez
s’avaient	aient	serais	vont
aurai	être	serait	se
m’aurai	suis	serions	me
auras	es	seriez	te nous vous en
t’auras	t’es	seraient	
aura	est	fus	

FIGURE A.1 – Liste des auxiliaires et des pronoms

Entrée	Lexs.	Score	Candidat	Syn
furieux#I	furax	0.926161	alcoolique	0
linge#I.b	petit linge	0.901918	chaussure	0
ouragan#II.3	avalanche	0.866089	torrent	1
grille-pain	toasteur	0.981849	sécurité	0
gamberger	réfléchir	0.916856	!	0
cricoïdien	crico-	0.961005	articulaire	0
heuchère	désespoir des peintres	0.900478	branche	0
dégueu#VI.3	dégueulasse	0.898663	énorme	0
bicyclette#I	biclou, biclo, bécane, vélocipède, vélo, petite reine	0.989853	voiture	0
lolos	seins, nénés, tétons, nichons, roploplos, roberts, nibards, mamelles	0.875500	bras	0
bol_2#I	chance, pot, chatte, cul	0.850563	problème	0
égarer#2	paumer, perdre	0.967881	lire	0
téloche#I	télévision, télé	0.898510	maison	0
gymnastique_N#I.1a	gym	0.935540	musculation	0
pro	professionnel	0.998830	dieu	0
encadrer#I.1b	encadrer	0.980271	remplir	0
rouge#VI.3a	communiste, coco	0.856143	socialiste	1
nettoyer#IV.2b	nettoyer	0.937918	réparer	0
cinéma#III	cinoche, comédie	0.893622	défi	0
net_Adj#III.2	précis, clair, distinct	0.943371	efficace	0
hôtesse	hôte	0.862829	ami	0
dent_1#I.a	quenotte, ratiche	0.953901	ride	0
craspec#III	cradoque, crado, cracra, cradingue, craspec, crade	0.859553	abouti	0
crade#V.2	craspec	0.857131	ou	0
fesses#I.b	arrière-train	0.870487	côté	0
chiffre#II.3	service du chiffre	0.893918	cnrs	0
thorax#a	cage thoracique	0.862348	cœur	0
mairie#1	hôtel de ville	0.981045	boulangerie	0
seins#I.2b	lolos, nénés, tétons, nichons, roploplos, roberts, nibards, mamelles	0.895673	pied	0
dégueu#VI.1	dégueulasse	0.946487	ça	0
cou#II	goulot, col	0.968327	fond	0
cinéma#I.2	ciné, septième art, cinoche	0.957890	théâtre	0
éducatrice#2	éducs	0.865020	garde	0
avocat_1#I	membre du barreau	0.859471	artiste	0
venger#I	venger	0.972504	rejoindre	0
craspec#III	cradoque, crado, cracra, cradingue, craspec, crade	0.972414	etc	0
veste_2#1	gamelle	0.942515	élection	0
boulot	travail, emploi	0.873927	blog	0
métro#I	métropolitain	0.981619	tramway	0
car_Conj#1	par conséquent, en conséquence, donc, alors	0.943670	et	0
kilogramme	kilo	0.987732	sorte	0
lombric	ver de terre	0.875923	ver	0
spam#I	pourriel	0.912196	virus	0
dégueu#V.1	dégueulasse	0.883218	mal	0
crasseux#II	cradoque, crado, cracra, cradingue, craspec, crade	0.914037	gay	0
sms#II	texto	0.887763	post	0
amarante#II	pourpre	0.975162	blanche	0
tante#I	tata, tatie, tantine	0.960406	cousin	0
survêt	jogging, survêtement	0.859401	gens	0
se déplacer#I.2d	attirer, déplacer	0.964144	retrouver	0
marcher#VII	rouler, tourner	0.936760	paraître	0
s'inquiéter	inquiéter	0.854779	rigoler	0
tandem#II	binôme	0.929924	commission	0
meurtre	homicide	0.886529	atteinte	0
manger_V#II.2	bouffer, se bouffer	0.909018	avoir	0
jam	bœuf, jam-session	0.959069	j	0
suspendre#II	interrompre	0.919017	arrêter	1
encadrer#IV.b	encadrer	0.927370	représenter	0
tante#I	tata, tatie, tantine	0.933789	soeur	0
angélique#2	d'ange	0.984235	même	0
nettoyer#I.1a	nettoyer	0.852211	défoncer	0
cigare#III.2	ciboulot, tête, cerveau, cafetière, matière grise, cervelle, caboche, crâne	0.913773	sourire	0
gorge#I.1	kiki, gaviot, gargamelle	0.895266	cou	0
au-dessous	au-dessus, en dessus, en dessous	0.909342	autour	0
crado#III	cradoque, cracra, crasseux, cradingue, crade	0.856668	pâle	0
tata#I	tantine, tante, tatie	0.957010	et	0
alcool#III.2	alcoolisme, boisson	0.863925	enfer	0
triche	tricherie	0.939328	jalousie	0

numéro#I.2	nº	0.864350	hôtel	0
se bouffer#IV.3	bouffer, manger	0.863948	donner	0
encadrer#II.1b	encadrer	0.895281	identifier	0
rebond#I	rebondissement	0.864946	doigt	0
cradoque#I.1	crado, cracra, crasseux, cradingue, craspec, crade	0.992970	!	0
rouge#III.3a	carton rouge	0.951459	cuir	0
judokate	judoka	0.869438	femme	0
colvert	malard	0.937090	lui	0
tempête#II.1	ouragan	0.935901	attaque	0
lac#II.1b	sinus, citerne	0.966094	filet	0
rouge#VI.2	communiste, coco	0.934408	chinois	0
toilettes#III.1a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.967908	lavabo	0
saucisson	sauciflard	0.962543	tomate	0
instrument#2	instrument de musique	0.988353	autres	0
dormir#I.1a	en écraser, roupiller, pieuter, ronfler, pioncer	0.854910	venir	0
liaison#II	liaison chimique	0.893666	force	0
78	soixante-dix huit	0.896693	53	0
papi#I	bon-papa, bonne-maman, mémère, mère-grand, grand-mère, mémé, grand-papa, grand-maman, pépé, mamie, aïeule, grand-père, ancêtre, aïeul, pépère	0.969592	père	0
promis_N	fiancé	0.903812	époux	0
gamberger	réfléchir	0.933301	;	0
crade#VI.3	cradoque, crado, cracra, crasseux, cradingue, craspec	0.901101	fait	0
nettoyer#IV.2a	nettoyer	0.960223	supprimer	0
narine#2	évent	0.865090	os	0
arrêter#V.2a	stopper	0.860287	avancer	0
désespérer#3	se désespérer, désespérer	0.850593	pour	0
rebond#I	rebondissement	0.938573	choc	0
bien_Adv#1	ben	0.867124	moins	0
dormir#I.1b	pioncer	0.868867	courir	0
bouger#I.2c	grouiller	0.904309	craquer	0
tabac_1#III	débit de tabac, bureau de tabac	0.912202	bout	0
manger_V#VI.1a	bouffer, se manger	0.941872	couvrir	0
round	reprise	0.897391	jour	0
âne#I	bourrique, baudet, bourriquet, bourricot	0.973563	marche	0
lion#I	roi des animaux	0.968076	ours	0
autocar	car	0.876386	ascenseur	0
gorge#I.1	kiki, gaviot, gargamelle	0.883196	bouche	0
phobie#2	-phobie	0.985278	complexe	0
auto-stop	stop	0.948925	voiture	0
jadis	autrefois, antan	0.916968	grâce	0
guitare	gratte	0.879728	voix	0
réveiller#a	éveiller	0.876214	chercher	0
crocodile#II	croco	0.898787	rouge	0
chanter#III.1	chanter	0.908831	avoir	0
crime#I.a	acte criminel	0.887270	peine	0
tête#I.1a	poire, crâne, chef, boule, pomme, caillou, cigare, carafe, caboche	0.870671	rue	0
chatte#III.1	moule, con, foufoune, vulve, zézette, minou, abricot	0.935859	femme	0
situé	sis	0.923936	caché	0
autobus	bus	0.920059	équipe	0
collaborateur_N#II	collabo	0.870119	terroriste	0
survêtement	jogging, survêt	0.999103	uniforme	0
stomacal	gastrique	0.897198	biologique	0
numéro#I.1	nº	0.865692	tronc	0
pesée	pesage	0.956950	préparation	0
treize#I.1a	xiii	0.883445	deux	0
niquer	baiser	0.884506	regarder	0
ample	large	0.907510	original	0
coiffeuse_1	coupe-tif	0.956454	voisin	0
bicyclette#I	biclou, biclo, bécane, vélocipède, vélo, petite reine	0.971100	pièce	0
soleil#I.a	astre du jour	0.914505	manteau	0
communiste_Adj#2	rouge, coco	0.875897	intelligent	0
cradingue#IV.2	cradoque, crado, cracra, crasseux, cradingue, craspec, crade	0.995438	vulgaire	0
distributeur#II.2	guichet	0.950834	homme	0
oncle	tonton	0.853186	père	0
furibard	furibond	0.918753	discret	0
bouée#III	brioche, pneu	0.859729	nourriture	0
commencer#I.2b	ouvrir, commencer, s'ouvrir	0.897603	contenir	0
gendarmette	gendarme	0.912485	brigade	0
cradoque#IV	crado, cracra, cradingue, craspec, crade	0.943010	décevant	0
baiser_V#II	niquer	0.928870	le	0

cigare#III.1	tête, poire, chef, boule, pomme, caillou, caboche, carafe, crâne	0.850129	fond	0
dégueulasse#II.2	dégueu	0.918357	riche	0
toilettes#III.1a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.964916	aire	0
3#2	trois	0.978409	12	0
Japonaise	nippone	0.956031	infirmier	0
boucher_N#III	charcutier	0.864914	homme	0
-phobie#2	phobie	0.854174	non	0
gamin_N#I.1	gosse, enfant	0.947096	gars	0
chatte#III.1	moule, con, foufoune, vulve, zézette, minou, abricot	0.938541	homme	0
perfection	parfait	0.981501	puissance	0
voiture#3	wagon	0.980882	voie	0
TER	train express régional	0.984652	bus	0
pudeur	pudicité	0.991739	fille	0
bouger#I.2c	grouiller	0.929352	sourire	0
dégueulasse#I.2	dégueu	0.983985	beau	0
cigare#III.2	ciboulot, tête, cerveau, cafetière, matière grise, cervelle, caboche, crâne	0.864141	talent	0
aube_1#I	pointe du jour	0.916347	heure	0
yaourt#a	yogourt	0.962324	fruit	0
thorax#a	cage thoracique	0.936418	mal	0
thorax#a	cage thoracique	0.941864	cou	0
téléphone#II.2	nº, numéro	0.950850	nom	0
orteil#2	pouce	0.899888	front	0
football#1	soccer, foot, ballon rond	0.922277	jeu	0
radiodiffusion	radio	0.894451	carbone	0
liaison#II	liaison chimique	0.881474	énergie	0
comprendre_2	entrer	0.990620	regrouper	0
se déplacer#I.2d	attirer, déplacer	0.882957	manifester	0
odorat	nez, olfaction	0.870734	coeur	0
cailler#I.2a	cailler	0.880767	faire	0
musique#III	rengaine, refrain, chanson	0.913214	chose	0
fesse#III	cul	0.910117	caméra	0
dormir#I.2	pioncer	0.933989	rester	0
cradingue#IV.2	cradoque, crado, cracra, crasseux, cradingue, craspec, crade	0.999591	habituel	0
crocodile#II	croco	0.866940	bleu	0
cheval#I.1a	dada	0.961449	système	0
tante#I	tata, tatie, tantine	0.940376	mère	0
pioncer#I.1a	en écraser, roupiller, pieuter, dormir, ronfler	0.924978	marcher	0
pneu#I	pneumatique	0.939467	phare	0
condom	capote, préservatif	0.942179	mec	0
gorge#I.1	kiki, gaviot, gargamelle	0.991877	nuque	0
jogging#II	survêt, survêtement	0.955239	costume	0
mécanicien_N#II.1	mécano	0.898647	jeune	0
gaillet	caille-lait	0.905237	thym	0
costume#I.2	costard	0.966223	pantalon	0
bouffer_1#I.3	se bouffer, manger	0.898614	perdre	0
cracra#I.1	cradoque, crado, crasseux, cradingue, craspec, crade	0.976358	petit	0
plat_Adj#I	aplati	0.963104	rouge	0
poitrine#4	coffre	0.915326	spectateur	0
stop#II.2	auto-stop	0.900840	vélo	0
minou#II	moule, con, foufoune, vulve, zézette, chatte, abricot	0.877068	cul	0
tramway#II	tram	0.853178	site	0
net_Adj#III.2	précis, clair, distinct	0.965514	varié	0
occident	ouest, ponant	0.889450	france	0
cadavre#I.a	cadavre, macchabée, macchab, dépouille	0.931489	président	0
porc#I.2	cochon	0.986049	lapin	0
nuage#I	nues	0.946266	bruit	0
tramway#I	tram	0.884361	train	0
pub	publicité	0.872736	émission	0
craspec#V.3	cradoque, crado, cracra, crasseux, cradingue, crade	0.966465	frais	0
dégueu#IV.1	dégueulasse	0.859937	mauvais	0
s'affoler	affoler	0.923568	oublier	0
s'éveiller	se réveiller	0.946422	rentrer	0
rock_N#a	rock'n'roll	0.880397	métal	0
kilogramme	kilo	0.906445	variété	0
parterre#II.2	champ	0.886048	troupeau	0
rhino	rhinocéros	0.867783	tigre	0
caille-lait	gaillet	0.990887	thym	0
zoo	jardin zoologique, parc zoologique	0.939539	site	0
léguer	hériter	0.869667	donner	0
manger_V#VII	bouffer	0.853677	rappeler	0

tornade#IV	ouragan	0.957309	silhouette	0
manger_V#VI.1b	bouffer, se bouffer, se manger	0.898148	traverser	0
alentours	environs	0.853151	mur	0
yaourt#a	yogourt	0.962567	miel	0
crasseux#I	cradoque, crado, cracra, cradingue, craspec, crade	0.953146	noir	0
bus	autobus	0.952347	anniversaire	0
avancée#I	avance	0.888179	autorité	0
nez#I.a	pif, nase, blair, tarin, naseaux, blase	0.865946	visage	0
boire_V#I.2b	picoler	0.882378	écrire	0
PCG	plan comptable général	0.919805	p	0
AVC	accident vasculaire cérébral	0.921821	patient	0
cheval#I.1a	dada	0.905940	anneau	0
rugby#1	ballon ovale	0.942467	basket	0
cracra#III	cradoque, crado, crasseux, cradingue, crade	0.942589	clair	0
-phobie#2	phobie	0.936101	ration	0
lever_V#VIII	débusquer	0.887746	chasser	0
cafard#I	blatte	0.883967	fourmi	0
SF	science-fiction	0.978629	littérature	0
s'agir#2	devoir	0.942026	suffire	0
cigare#V	colombin	0.924836	truc	0
cailler#I.2b	cailler	0.984755	transformer	0
chiffre#II.2	combinaison	0.882755	bagage	0
cigare#IV	phallus	0.984858	cul	0
nez#I.a	pif, nase, blair, tarin, naseaux, blase	0.902044	salon	0
couvert_Adj#I.2	recouvert	0.939010	long	0
succinct#II	sommaire	0.871619	extérieur	0
mère_1#I.1a	daronne, mater	0.857313	adulte	0
crachat	glaviot	0.904001	chien	0
fesse#III	cul	0.932770	plomb	0
zézette	moule, con, foufoune, vulve, chatte, minou, abricot	0.970757	tête	0
cadavre#I.b	cadavre, macchabée, macchab, dépouille	0.937902	corps	1
pousser#VII.1	lever	0.962229	porter	0
machine#2	lave-linge	0.936603	servant	0
fessier_N#2	cul, séant, paire de fesses, postérieur, derrière, popotin, lune, derche, miches, pétard, croupion, arrière-train, croupe, panier	0.939537	vincennes	0
autobus	bus	0.979174	homme	0
crico-	cricoïdien	0.885861	intra	0
cinoche#I	cinéma, septième art, ciné	0.977406	télévision	0
voiture#3	wagon	0.894992	locomotive	0
situé	sis	0.969081	sur	0
sensible#I.2	sensible	0.951508	provenant	0
rajeunir#2	rajeunir	0.958775	être	0
cure-pipe	nettoie-pipe	0.955045	cou	0
bouffer_1#I.2	manger	0.961495	dévorer	1

TABLEAU A.3 – Évaluation des candidats absents du DS 0 avec un score normalisé ≥ 0,85 < 1 (méthode de base)

Entrée	**Lexs.**	**Score**	**Candidat**	**Syn**
proche_Adj#IV.1	voisin	0.856506	canon	0
crasseux#II	cradoque, sale, crado, cracra, dégueu, dégueulasse, cradingue, craspec, crade	0.914037	gay	0
net_Adj#III.2	précis, clair, détaillé, distinct	0.918274	facile	0
dessous_N#II	aspect	0.926952	étape	0
vingt#1b	vingtaine	0.927426	cinquante	0
boule#I.2a	peloton, sphère, cristal, globe	0.912268	boîte	0
cradoque#II	sale, crado, cracra, dégueu, dégueulasse, crasseux, cradingue, craspec, crade	0.954125	moche	0
atermoyer	tergiverser, noyer le poisson, botter en touche	0.857771	le	0
visage#II.1	caractère, personnalité	0.930929	professeur	0
éloigné#I	distant, à l'écart, lointain	0.868219	proche	0
fanfare#I.2	zique, zizique, musique	0.931383	étoile	0
liaison#II	liaison chimique	0.964696	molécule	0
plat_Adj#I	ras, aplati	0.964143	blanc	0
dévoiler#II	vendre la mèche, révéler, découvrir	0.962669	exposer	1
taxé	imposable	0.855473	traité	0
boire_V#II	flamber, dépenser	0.925638	pour	0
intuition#I	nez, instinct	0.898258	histoire	0
gonfler#II	se gonfler, lever	0.905005	passer	0

karaté	lutte, kung-fu, taekwondo, aïkido, jujitsu, judo	0.853252	rue	0
tornade#V	tempête, torrent	0.913077	vingtaine	0
football#1	soccer, foot, ballon rond	0.922277	jeu	1
meute#1	horde	0.889502	quarantaine	0
vivacité#IV	rapidité	0.851227	intensité	0
photographie#II	photo, tirage	0.956922	tête	0
gonfler#II	se gonfler, lever	0.919325	rougir	0
coffre-fort#a	armoire forte	0.905087	wc	0
obstiné_Adj#a	cabochard, coriace, têtu	0.962538	seul	0
marché#I	accord	0.905327	contrat	1
cracher#III.1	lancer	0.881757	lâcher	1
mélodie#I.2	thème	0.935752	symphonie	0
beurrer	tartiner	0.875082	faire	0
maillot#I.1	maillot vert, maillot de corps, tee-shirt	0.871387	pantalon	0
nez#IV	avant, museau, proue	0.915994	long	0
partir_1#I.1a	prendre la route, s'en aller, repartir, quitter, larguer les amarres, se retirer, s'envoler	0.947740	travailler	0
soutien-gorge	dessous, soutif, sous-vêtement	0.934274	plafond	0
piano#I	piano droit, piano à queue	0.898800	christ	0
dégueu#VI.1	cradoque, sale, crado, cracra, dégueulasse, salaud, cradingue, craspec, crade	0.946487	ça	0
éduquer#a	élever	0.868068	traiter	0
orphelin_Adj#I.1a	orphelin, pupille, abandonné	0.864468	gâté	0
gymnastique_N#I.2	gym, séance	0.926583	apparition	0
cuisines#II.b	cuisine roulante, office, cuisine	0.850122	chaise	0
aimantin	magnet, aimant	0.871220	logo	0
sapin#I.b	arbre de noël, sapin	0.901763	lit	0
insecte#II	vermine	0.888220	femme	0
trébucher_1	buter	0.967491	coucher	0
vote#I	voix, suffrage	0.970489	film	0
antenne#II.1	récepteur, émetteur, capteur	0.879682	arme	0
riche_Adj#II.2	goûteux, abondant	0.991197	décoré	0
paire#2	couple	0.949184	douzaine	0
tramway#I	tram	0.884361	train	1
torse#1	cage thoracique, thorax, poitrine	0.919307	look	0
sphère#I.1	boule	0.850178	bouteille	0
route#IV	chemin, marche, progression, course, voie	0.948646	remontée	0
nuageux#2	couvert	0.888899	ancien	0
seins#I.2b	lolos, seins, poitrine, sein gauche, nénés, tétons, nichons, roploplos, roberts, sein droit, nibards, mamelles	0.979227	homme	0
train_1#I	métro, train express régional, corail, ter, tortillard, express, rapide, train à grande vitesse, omnibus, r.e.r, t.g.v.	0.918249	avion	0
inexactitude	fausseté	0.941451	absence	0
vague_Adj#II.2	lointain	0.851298	vif	0
rapide_Adj#I.2	preste, leste, vif	0.883260	léger	1
timide_Adj#II	faible	0.935847	intime	0
toasté	grillé	0.889508	généreusement	0
énervé	fâché, hors de ses gonds, agacé, le couteau entre les dents	0.865501	non	0
malicieux	espiègle, gamin	0.948579	désespéré	0
table#II.2	tablée	0.991768	salle	1
exercer#II.2	bosser, travailler, officier	0.856130	rester	0
gouttière#I	chéneau	0.943769	maison	0
brosse#I.2	pinceau	0.900172	magazine	0
liste	catalogue, répertoire, table	0.957304	dizaine	0
nuageux#2	couvert	0.867533	géologique	0
mère_1#IV.1b	principal	0.942879	centrale	1
ennemi_N#II	misanthrope, adversaire, misogyne, opposant	0.977125	fondateur	0
bouleverser#II	bousculer, chambouler	0.997805	influencer	0
gain#I	bénéfice, revenu, surplus, bic, bénéfices industriels et commerciaux	0.950857	investisseur	0
maisonnette	maison	0.975058	symphonie	0
champ_1#II.2	parterre, étendue	0.947640	paysage	0
se chipoter	se quereller, s'engueuler, se disputer	0.876270	chercher	0
hold-up#II	attaque, vol, appropriation	0.927925	documentaire	0
réduction#III	canapés, toast, amuse-gueule, amuse-bouche, petit four	0.987135	friandise	1
effarant	affolant	0.961617	intéressant	0
avancée#III	promontoire	0.862394	plage	0
serré#I	compact, écrasé	0.995214	glissé	0
battre_1#IV.1	gagner	0.885692	vaincre	1
se raccommoder	se réconcilier	0.890758	retrouver	0
contorsion#I	gymnastique	0.979207	concentration	0
informatique_Adj	informatisé, électronique	0.881996	indispensable	0
alcool#III.1	bouteille	0.901883	ogm	0
boucher_V#I.1b	obstruer	0.958587	percer	0
craintif#1b	peureux, couard	0.964012	soumettre	0
route#II.1	trajet, chemin, parcours, itinéraire	0.990579	liaison	0

convoitise	envie	0.863470	nécessité	0
toilette#I.1a	soin	0.860381	douche	0
terrible#III	sévère	0.951630	exemplaire	0
gendarmette	chaussettes à clous, gendarme	0.881684	compagnie	0
objectif_Adj	impartial	0.885731	public	0
comme#IV	presque	0.884921	toujours	0
orpheline#a	pupille, abandonnée, orpheline	0.885132	enfant	1
concert#I	récital	0.938872	art	0
lunch	déj, déjeuner, repas	0.914654	travail	0
football#3	foot, partie, match	0.894961	stage	0
commémoration	célébration	0.917615	soirée	0
peut-être	possiblement	0.890323	certainement	0
manger_V#I.2	détériorer, bouffer	0.975218	aspirer	0
téléphone#II.2	nº, numéro	0.950850	nom	0
avancée#III	promontoire	0.913422	île	0
ciné#I	grand écran, cinéma, septième art, cinoche	0.978911	vin	0
éloignement#I	écartement	0.888027	hygiène	0
marcher#VII	réussir, tenir la route, aller, tourner, rouler, fonctionner, se tenir	0.936760	paraître	0
généreusement	charitablement	0.901890	gratuitement	0
allonger_1#II.1	agrandir, rallonger	0.938861	sauver	0
baiser_V#I	embrasser	0.912450	casser	0
abondant#I.2	riche	0.864594	tout	0
onirique#2	irréel	0.852204	différent	0
nuage#II.4	amas	0.982352	million	0
infirme_Adj	handicapé	0.856942	malade	1
exagérer#III	en faire des tartines, pousser le bouchon trop loin, en rajouter, tartiner, en mettre des tartines, pousser, charrier	0.904652	rêver	0
s'effondrer#II	crouler, s'écrouler	0.954314	pencher	0
nase_1#I	trou de nez, pif, narines, naseau, nez, blair, museau, tarin, naseaux, blase	0.886649	sac	0
objet#I.1	excentricité, bidule, rareté, truc, chose, gadget, poids, machin, pièce	0.957444	souvenir	0
sommeiller#II.a	dormir, roupiller, sommeiller	0.911158	progresser	0
train_1#V.1	succession, rythme, série, suite	0.956824	ligne	1
s'allonger_1#II.1	grandir	0.976425	croître	1
fond#I	poche, paroi, cul	0.984987	coffre	0
adresse_1#I	habileté, coordonnées	0.970118	identité	0
croustillant#I	croquant	0.930400	fait	0
bazar#I	marché, souk	0.974958	voile	0
conjointement	collectivement	0.904409	annuellement	0
bureau#I	secrétaire, table de travail	0.875235	sol	0
lessive#IV	coup de balai, nettoyage, tri	0.899369	accumulation	0
bord#I.2	côté, rive, plage	0.942604	site	0
manuel_Adj#1	à la main, à main	0.875481	complet	0
inaudible	non audible, imperceptible	0.851488	doux	0
blague_1#1	jeu de mots, calembour	0.932791	truc	0
rive#I.1a	coteau, bord, rive, rivage, berge	0.994082	maison	0
se déplacer#I.2d	accrocher, venir, déplacer, aspirer, attirer, courir	0.982515	manifester	0
trésorerie#I.1	administration	0.993564	direction	0
éduqué#b	propre	0.863612	affectueux	0
incroyable#I	fou, à dormir debout	0.908203	historique	0
air_2	apparence, figure	0.911462	visage	1
bouffer_1#IV.4b	critiquer, manger du curé	0.866721	parler	0
poule_1#II	cocotte	0.960151	femme	1
thorax#a	poitrine, cage thoracique, coffre, buste, torse	0.941864	cou	0
formulaire	imprimé	0.983711	document	1
museau#I.1a	nez, groin	0.859352	bec	1
douche#IV	averse	0.966348	diversité	0
se cailler#II	avoir froid, se geler les fesses, geler, cailler	0.940153	faire	0
3#2	trois	0.968322	2	0
manufacture	usine	0.932311	boutique	0
callipyge_Adj	fessu	0.868336	malade	0
tempête#II.1	bouleversement, tornade, agitation, tonnerre, difficulté, troubles, ouragan, orage, tourmente	0.935901	attaque	0
fesses#I.a	siège, cul, paire de fesses, séant, postérieur, derrière, popotin, derche, lune, miches, pétard, fessier, croupion, arrière-train, croupe, ballons, panier	0.976120	diamant	0
coiffeuse_2	table de toilette	0.895756	fauteuil	0
sportive#I	licenciée, moustique, poussins	0.966972	sportif	0
aborigène_Adj	indigène	0.872651	personnel	0
étouffé#I	asphyxié	0.955084	blessé	0
calme_Adj#I	tranquille	0.892692	sérieux	0
ampoule#II	cloque	0.859666	poil	0
flegmatique#a	lymphatique	0.944476	agressif	0
couleur#II.1	colorant, teinture	0.986385	chose	0
clouer#I	river, fixer	0.922600	installer	1

bandit#II.2	garnement, vaurien, filou, crapule, chenapan, canaille, voyou	0.937291	gourmand	0
bouilloire	samovar	0.978337	poêle	0
cheptel	troupeau, bétail	0.953029	nombre	0
effort#I	coup de collier, action	0.947733	élan	1
franchissable	traversable, guéable	0.864095	visible	0
fauteuil#II	fonction, siège, poste	0.982538	verre	0
tee-shirt	maillot	0.939637	coton	0
écarter#I.2	repousser, chasser	0.994157	trouver	0
attendre	patienter	0.856787	passer	0
similitude	analogie, point commun, ressemblance, similarité, identité	0.985493	situation	0
fesses#I.a	siège, cul, paire de fesses, séant, postérieur, derrière, popotin, derche, lune, miches, pétard, fessier, croupion, arrière-train, croupe, ballons, panier	0.888837	jambe	0
éléphant#II.2	ponte, ténor	0.886630	ordre	0
orphelin_Adj#II.1	veuf, sans	0.956822	amputé	1
acide_Adj#1	piquant	0.877630	sucré	0
lac#II.2	océan, champ, étendue, mer	0.946333	tapis	0
s'engager#I.1	entrer, rejoindre	0.904614	monter	0
se reproduire#I	se produire	0.870891	produire	0
environnement#I	cadre, milieu	0.885473	enceinte	0
foudre_1#II.1	accusation	0.886464	arme	0
sportif_N	licencié, moustique, poussins	0.965079	enfant	0
éclair_1#I.2	lueur	0.899832	onde	0
grandeur#I	long, longueur, taille	0.985740	disposition	0
ancien_Adj#I.1a	vieux	0.879867	lumineux	0
cigare#II	tabac	0.985321	café	0
grille-pain	toasteur	0.945122	paquet	0
vache#II.1b	gros, grosse, baleine, éléphant	0.927835	moto	0
laver#VI	blanchir, venger	0.956337	libérer	0
éducatrice#2	éducs	0.865020	garde	0
entrer#II.1	rejoindre, s'engager, engager	0.904518	être	0
tenir#VI.1b	continuer, résister, se maintenir	0.857721	être	0
transaction	opération, opa, offre publique d'achat, échange, ope, offre publique d'échange, achat	0.960374	somme	0
franc_Adj_1#I.2a	honnête, sincère	0.975250	maladroit	0
chanter#III.2	chanter	0.974210	évoquer	0
gueule#I.2a	bouche	0.964956	pluie	0
violon#II	prison	0.919453	lit	0
mouche#II.1	leurre, appât	0.903311	matériel	0
s'éloigner#I.1	s'écarter	0.865181	plaire	0
manteau#V.1	couverture, revêtement, habit, vêtement	0.948142	paysage	0
explosion#II	coup de grisou, tonnerre	0.856547	crise	1
écrire#II	pondre, gribouiller, noter, rédiger	0.877032	dire	0
concombre#b	cornichon	0.947615	carotte	0
table#II.2	tablée	0.910166	planète	0
manger_V#VI.2a	bouffer, avaler	0.965784	refuser	0
zézette	moule, tête à claques, con, foufoune, sexe, vulve, chatte, minou, abricot	0.970757	tête	0
tige#II	baguette, jambe	0.900108	plaque	0
raté	mouton noir, canard boiteux	0.911349	chien	0
coup#V.1	coup dur	0.996360	chagrin	0
impliquer#II.2	atteindre	0.881029	élever	0
index#II.1	sommaire	0.952457	histoire	0
jusque#I	à concurrence	0.987335	et	0
dégueu#VI.3	cradoque, sale, crado, dégueulasse, crasseux, crade	0.982805	sauvage	0
cou#II	goulot, col	0.968327	fond	0
orphelin_Adj#II.3	dépourvu, sans	0.950527	recouvert	0
conduire#I.3	mener	0.985213	situer	0
pluie#II.2	bordée, averse, déluge, avalanche, flot, grêle, torrent, ouragan	0.921530	trio	0
raccourcir#I	réduire, diminuer	0.963915	doubler	0
appliqué#II	soigné	0.868394	engagé	0
niquer	faire son quatre-heures, faire l'amour, dormir, baiser, tirer un coup	0.884506	regarder	0
appartement#1	garçonnière, loft, habitation, quadruplex, duplex, penthouse, meublé, clapier, triplex, studio, appart	0.988785	bâtiment	1
gentiment	aimablement	0.851069	doucement	1
narine#2	trou de nez, narines, évent, conduit, orifice	0.865090	os	0
phobie#2	allergie, -phobie, appréhension, aversion	0.985278	complexe	0
barbiche#2	barbe	0.899096	main	0
ici#I.a	près, partout, loin, ailleurs, quelque part, là-bas, là, y, par là	0.918353	humble	0
unicité	unité	0.853085	existence	0
alcool#IV	drogue	0.950283	sentiment	0
appliqué#I	placé	0.863749	fixé	1
drôlement#II	très	0.851213	assez	1
bataille#II.1	confrontation	0.986344	relation	0
désert_N#X-IV.1	placard, purgatoire, infortune, marasme, passage à vide	0.871949	tunnel	0

violon#I.b	violoniste	0.984196	violoncelle	1
siester	en écraser, roupiller, pieuter, dormir, ronfler, pioncer	0.912883	nous	0
barque	bateau	0.917763	gare	0
en_Prép#IV.1	au moyen, avec, à travers, à bord, par	0.904100	à	1
partage#I	liquidation	0.977739	contrôle	0
apaiser#1	calmer	0.862725	protéger	0
pipe#V	conduit, tuyau	0.873683	sortie	0
tapis#III	parterre, couche, champ	0.886659	plat	0
voyou_N#1	criminel, malfaiteur, délinquant, racaille, arsouille, malfrat, loubard, bandit, gangster	0.933478	tueur	0
silhouette#a	formes	0.949397	robe	0
incontestable	indiscutable	0.971841	évident	1
vague_Adj#II.2	lointain	0.880543	brillant	0
marcher#II	avancer, progresser	0.888703	défiler	1
bêtement#1a	stupidement	0.860652	vite	0
habit#III	couverture, apparence, manteau, revêtement, vêtement	0.977991	salade	0
tinter	cliqueter	0.853615	glisser	0
muscle#I.1a	muscle, myocarde, muscles	0.989389	doigt	0
fiable	sûr	0.926276	connu	0
ventilation#II	répartition	0.967898	réalisation	0
bien-aimée	amour, objet aimé	0.968854	femme	0
antan	jadis, hier, dans le temps, autrefois	0.911178	espagne	0

TABLEAU A.4 – Évaluation des candidats absents du DS 1 avec un score normalisé ≥ 0,85 < 1 (méthode de base)

Entrée	**Lexs.**	**Score**	**Candidat**	**Qsyn**
bouffer_1#IV.3	se bouffer, manger	1	parler	0
bouffer_1#VI.3	se bouffer, se manger, manger	1	passer	0
dégueu#VI.4	dégueulasse	1	quoi	0
arriver#I.2	livrer	1	tomber	0
rouge#III.3a	carton rouge	1	carton	0
étouffer#II.2a	étouffer	1	accompagner	0
ivre#I.1	casquette	1	!	0
mari	époux	1	oncle	0
métro#I	métropolitain	1	désert	0
cracra#II	cradoque, crado, crasseux, cradingue, craspec, crade	1	!	0
orteils#1a	doigt de pied	1	bras	0
apprendre#I.2a	apprendre	1	arrêter	0
couleur#II.2b	coloration, teinture	1	robe	0
cradingue#III	cradoque, crado, cracra, crasseux, crade	1	fluo	0
père#I.1a	pater, daron, dab, paternel	1	acteur	0
longueur#V	saut en longueur	1	guerre	0
pompes_2	tatanes, grolles, chaussures, godasses	1	chausson	0
bifteck#I	steak	1	bonbon	0
musicien_N	musico	1	femme	0
vrai_Adj#III	véritable	1	grand	0
mayonnaise	mayo	1	moutarde	0
bouger#I.2c	grouiller	1	cesser	0
fragiliser	se fragiliser	1	endommager	0
bol_1#I.b	bolée	1	verre	0
vêtement#II	fringue	1	textile	0
éveil#I.1	réveil	1	autre	0
mayonnaise	mayo	1	salade	0
cinéma#III	cinoche, comédie	1	discours	0
binôme#II.1	tandem	1	groupe	0
cerveau#I.2	ciboulot, tête, cafetière, matière grise, cervelle, caboche, cigare, crâne	1	monde	0
sale_Adj#V.1	salaud	1	ça	0
tornade#IV	ouragan	1	une	0
pères#II.2	ancêtres, aïeux	1	parent	0
courir#III.2	courre	1	chasser	1
fleurissement#1	floraison	1	ramassage	0
causant	causeur	1	sympa	0
seins#I.2b	lolos, nénés, tétons, nichons, roploplos, roberts, nibards, mamelles	1	yeux	0
dégueulasse#VI.3	dégueu	1	mort	0
dormir#III	reposer	1	mourir	0
1500	mille cinq cents	1	cent	0
caille-lait	gaillet	1	romarin	0

W.-C.#a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	...	0
tante#I	tata, tatie, tantine	1	voisin	0
faux_Adj#IV	erroné, déformé	1	autre	0
réveil#II	réveil-matin, réveil	1	renouveau	0
musique#I.2a	zique, musique, zizique	1	bruit	0
pesage	pesée	1	marquage	0
crade#II	cradoque, crado, cracra, crasseux, cradingue, craspec, crade	1	beau	0
marcher#VI.2	alimenter	1	de	0
dégueu#IV.1	dégueulasse	1	mauvais	1
rouge#VI.3b	communiste, coco	1	fille	0
linge#I.b	petit linge	1	tenue	0
crade#VI.2	crado, crasseux	1	répétitif	0
cracra#II	cradoque, crado, crasseux, cradingue, craspec, crade	1	:	0
conteuse#1	raconteuse	1	enfant	0
pourpre#I.1a	amarante	1	blanc	0
dégueulasse#VI.3	dégueu	1	ignoble	1
cradoque#I.2	cradingue, crado, crade, cracra	1	long	0
guitoune#1b	tente	1	chevaux	0
gnôle	eau-de-vie	1	vin	0
tinettes#b	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	passager	0
thorax#a	cage thoracique	1	ventre	0
étouffer#II.1b	étouffer	1	être	0
cul-de-sac	impasse	1	labyrinthe	0
rouge#IV	vin rouge	1	repas	0
se glisser	se fourrer	1	cacher	0
excréments	matière fécale, bol fécal, fèces, selle	1	déchet	1
aspirateur_N#I.1	aspi	1	générateur	0
poire#II	tête, crâne, chef, boule, pomme, caillou, cigare, carafe, caboche	1	figure	0
rock_N#a	rock'n'roll	1	musique	0
bureau#II.3a	cabinet	1	service	0
transat	transatlantique, chaise longue	1	sac	0
motocyclisme	moto	1	motard	0
tombeau	sépulcre	1	arbre	0
nettoyage#I.1a	nettoyage	1	frai	0
drosophile	mouche du vinaigre	1	mouche	0
enceinte_Adj	en cloque, grosse	1	normal	0
éveil#I.2	veille	1	effort	0
immédiatement	sur le champ, tout de suite	1	donc	0
cousin_1#II	voisin	1	ami	0
boire_V#I.2a	picoler	1	mais	0
avalanche#III	ouragan	1	journée	0
manger_V#I.2	bouffer	1	perdre	0
expression#III	formule	1	phrase	0
cheveux#I.2a	chevelure, tifs	1	pied	0
cocotier	coco	1	fleur	0
cradingue#I.1	cradoque, crado, cracra, crasseux, craspec, crade	1	sale	1
mesure_1#II	unité de mesure	1	différence	0
inhumer	enterrer	1	attaquer	0
lac#II.1b	sinus, citerne	1	compartiment	0
craspec#I	cradoque, crado, cracra, crasseux, cradingue, crade	1	sombre	0
retentir#I	retentir	1	arriver	0
par#II.1	à travers	1	pour	0
bol_1#I.b	bolée	1	verre	0
fèces	excréments, selle, bol fécal, matière fécale	1	sang	0
CRC	chambre régionale des comptes	1	région	0
canapé#I	canap	1	fauteuil	0
colibri	oiseau-mouche	1	oiseau	0
Italien_N#I	macaroni	1	homme	0
brûler#II	griller	1	voir	0
homme#1a	humain	1	espèce	0
football#3	foot	1	tour	0
père#II.1a	dieu	1	grand	0
oxalide	oxalis, oseille sauvage	1	...	0
bosser	travailler	1	marcher	0
sommaire_Adj#II	succinct	1	irrégulier	0
valise#I	valoche	1	maison	0
venger#I	venger	1	retrouver	0
capote#II	condom, préservatif	1	porter	0
kidnappage	kidnapping	1	meurtre	0

dégueu#VI.4	dégueulasse	1	ça	0
bol_2#I	chance, pot, chatte, cul	1	respect	0
toilettage	toilette	1	brossage	0
s'intégrer#I	intégrer	1	convier	0
vieillarde	grand-mère, mamie, mémé	1	femme	0
dent_1#I.a	quenotte, ratiche	1	yeux	0
tabac_1#III	débit de tabac, bureau de tabac	1	marché	0
grand-mère#I	bon-papa, bonne-maman, mémère, mère-grand, mémé, grand-papa, grand-maman, pépère, pépé, mamie, aïeule, grand-père, ancêtre, aïeul, papi	1	mère	0
tante#I	tata, tatie, tantine	1	tant	0
cinéma#III	cinoche, comédie	1	conflit	0
réactif_N#1	réactant	1	matériaux	0
chuchotement	chuchotis	1	message	0
venger#I	venger	1	savoir	0
quasi#1	quasiment	1	en	0
téléphone#I	appareil téléphonique, bigophone	1	vase	0
pipe#III	pompier, turlute, fellation	1	affaire	0
skis#I.a	planches, lattes	1	chaussure	0
basket-ball#1	basket	1	football	0
bol_1#I.b	bolée	1	peu	0
narines#1	trou de nez	1	aile	0
téléphone#I	appareil téléphonique, bigophone	1	visage	0
chiffre#I.1b	nombre	1	nouveau	0
volcan#I	montagne de feu	1	montagne	0
e-mail#I	courriel, courrier électronique, mél, mail	1	message	0
s'enlaidir	enlaidir	1	changer	0
moto#II	motocyclisme	1	randonnée	0
gymnastique_N#I.1a	gym	1	sport	0
impeccable#I	impec	1	elle	0
s'arrêter#III.2	arrêter	1	cesser	1
mécanicien_N#II.1	mécano	1	constructeur	0
couleur#II.2a	coloration, teinture	1	foi	0
manger_V#I.4b	bouffer	1	prendre	0
purgatoire#II	désert	1	séjour	0
Vierge_N#2	mère de dieu, bonne mère, vierge mère	1	mère	0
dégueulasse#V.1	dégueu	1	serein	0
crade#VI.1	cradoque, crado, cracra, crasseux, cradingue, craspec, crade	1	vestimentaire	0
moto#I	bécane, motocyclette, meule	1	machine	0
narine#2	évent	1	sorte	0
Dieu#2	père	1	elle	0
rouge#VI.1	communiste, coco	1	politique	0
cigare#III.1	tête, poire, chef, boule, pomme, caillou, caboche, carafe, crâne	1	dos	0
mobile_N_2#I	portable, téléphone portable	1	téléphone	1
skis#I.a	planches, lattes	1	pied	0
alcoolique_Adj#2	alcoolo	1	fou	0
escargot#I.a	limaçon, colimaçon	1	lion	0
nuage#I	nues	1	blé	0
yaourt#b	yogourt	1	steak	0
locomotive	loco	1	voie	0
immortel_Adj#II	mémorable, inoubliable	1	de	0
inguérissable	incurable	1	grave	0
crado#V.2	crasseux, crade	1	moi	0
encadrer#I.1a	encadrer	1	fixer	0
cadavre#I.a	cadavre, macchabée, macchab, dépouille	1	corps	1
équitation	cheval	1	informatique	0
bouffetance	bouffe, boustifaille	1	semoule	0
sculptage	sculpture	1	jeu	0
manger_V#VI.1a	bouffer, se manger	1	et	0
crim'	brigade criminelle, criminelle	1	police	0
rouler_1#VII.1	tourner, marcher	1	commencer	0
une-pièce	t1, f1	1	coin	0
crade#V.2	craspec	1	etc	0
manger_V#VII	bouffer	1	confondre	0
chien#I.a	toutou, cador, cabot, clebs, clébard	1	animal	0
cinoche#I	cinéma, septième art, ciné	1	rock	0
ancêtres#II	pères, aïeux	1	origine	0
50	cinquante	1	10	0
connaître#III	toucher	1	vivre	0
batraciens#a	amphibiens	1	insecte	0
musique#III	rengaine, refrain, chanson	1	ambiance	0
folle#II	tapette, tante, tata, tantouze	1	star	0

tuer#II	décéder, mourir, emporter, périr, trépasser, disparaître, s'endormir, clamser, perdre la vie, casser sa pipe, crever, expirer	1	détruire	0
rouge#VI.3a	communiste, coco	1	conservateur	0
bonhomme	homme	1	Monsieur	1
con_N_2#II	moule, con, foufoune, vulve, zézette, chatte, minou, abricot	1	homme	0
patate#I.1	pomme de terre	1	poisson	0
se manger#III.a	bouffer, se bouffer, manger	1	donner	0
bouffer_1#III.1	se bouffer, manger	1	cacher	0
se bouffer#I	bouffer, se manger, manger	1	faire	0
rock_N#a	rock'n'roll	1	halloween	0
virus#III	maladie	1	plaisir	0
kiki	gaviot, gorge, gargamelle	1	cou	0
commodités#II.2a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	clé	0
couleur#II.2b	coloration, teinture	1	coupe	0
métro#I	métropolitain	1	centre	0
nettoyer#I.1b	nettoyer	1	toucher	0
cradingue#VI	cradoque, crado, cracra, craspec, crade	1	similaire	0
minou#II	moule, con, foufoune, vulve, zézette, chatte, abricot	1	gland	0
bouffer_1#VI.1	manger	1	bien	0
boire_V#I.2b	picoler	1	consommer	1
néerlandais_Adj#I.2	hollandais	1	français	0
manger_V#VI.1a	bouffer, se manger	1	couvrir	1
aube_1#I	pointe du jour	1	école	0
poitrine#4	coffre	1	respiration	0
roupiller#II.2	dormir, sommeiller	1	aussi	0
dégueu#V.2	dégueulasse	1	chaud	0
collabo_N#b	collaboratrice	1	patriote	0
manquer#I.1	manquer	1	profiter	0
dégueulasse#IV.2	dégueu	1	gras	0
s'endormir#III	décéder, mourir, emporter, périr, trépasser, disparaître, clamser, expirer, perdre la vie, casser sa pipe, crever, tuer	1	vivre	0
publicité	pub	1	musique	0
désintéresser	se désintéresser	1	nommer	0
rugby#1	ballon ovale	1	foot	0
aube_1#I	pointe du jour	1	heure	0
maladie#III.2b	virus	1	passion	1
champ_2	champagne	1	vent	0
rêve#I	songe	1	cas	0
apprendre#I.2b	apprendre	1	donner	0
yogourt#b	yaourt	1	biscuit	0
lever_V#IV.2	pousser	1	passer	0
véritable#III	vrai	1	très	0
boire_V#I.2a	picoler	1	raison	0
dégueulasse#VI.4	dégueu	1	beau	0
désert_N#X-IV.2	vide	1	problème	0
gorge#I.1	kiki, gaviot, gargamelle	1	bouche	0
fesses#I.b	arrière-train	1	main	0
s'appeler#II	être	1	signifier	0
canap	canapé	1	lit	0
cousin_1#II	voisin	1	congénère	0
W.-C.#b	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	chambre	0
cercueil#I	bière, sapin	1	armoire	0
brasse-papillon	papillon	1	natation	0
nullement	aucunement	1	non	1
amphibiens	batraciens	1	homme	0
bouffer_1#IV.1	se bouffer, manger	1	prendre	1
réactant	réactif	1	parasite	0
rock_N#a	rock'n'roll	1	genre	0
chat_1#I.b	félinés	1	mâle	0
boîte#II.1	entreprise	1	formation	0
manger_V#I.4b	bouffer	1	sucer	0
dégueulasse#VI.1	dégueu	1	bizarre	0
dégueu#V.1	dégueulasse	1	rien	0
défaire#I	se défaire	1	ouvrir	0
baiser_V#II	niquer	1	être	0
musique#I.1a	zique, musique, zizique	1	alarme	0
Nippon_N#I	japonais	1	frère	0
costard-cravate	costume-cravate	1	cravate	0
pétrin#II	caca, merde	1	piège	0
odorat	nez, olfaction	1	oeil	0
bicyclette#I	biclou, biclo, bécane, vélocipède, vélo, petite reine	1	voiture	0

joujou	jouet	1	livre	0

TABLEAU A.5 – Évaluation des candidats absents du DS 0 avec un score normalisé = 1 (méthode de base)

Entrée	Lexs.	Score	Candidat	Qsyn
ressemblance#2	parenté, air de famille, similitude	1	liaison	0
vents	instrument à vent	1	voix	0
bocal#I.1	conserve	1	sac	0
grand-mère#I	bon-papa, bonne-maman, mémère, mère-grand, mémé, grand-papa, grand-maman, pépère, pépé, grands-parents, mamie, aïeule, grand-père, ancêtre, aïeul, papi	1	vie	0
récepteur_N	oreille, antenne, capteur	1	prise	0
capturer	attraper	1	comme	0
sans#I.3b	sans	1	aussi	0
foyer#III.1b	hypocentre	1	cratère	0
emplir	bourrer, remplir, boucher, charger, combler	1	visiter	0
cordial	amical	1	poli	1
mélancolie#II.1	blues, déprime, spleen, tristesse, vague à l'âme, cafard	1	brume	0
faire#III.3b	nommer, promouvoir	1	rendre	1
mesure_1#I.2a	comptabilité	1	analyse	0
minutieux	détaillé, méticuleux, soigneux, approfondi	1	rigoureux	1
moustique#III.2	sportive, sportif	1	senior	0
volcan#III.1	bombe sexuelle	1	violent	0
préfecture#I	sous-préfecture	1	situation	0
s'entendre#1	se comprendre	1	communiquer	1
réveiller#a	éveiller	1	voir	0
orpheline#b	abandonnée, orpheline	1	femelle	0
correctement#2	bien, ben, convenablement, honnêtement, proprement	1	tranquillement	0
boucher_N#I.2	charcutier, boucher-charcutier	1	personne	0
crier#I.1a	hurler, braire	1	chanter	0
motif#II	imprimé	1	mur	0
toilette#VI	crépine	1	boîte	0
violon#I.b	violoniste	1	chanson	0
sortir#II.2	déployer	1	!	0
subdivision#2	partie, division	1	catégorie	1
emmerder	casser les couilles, faire chier, casser les pieds, énerver, faire braire	1	aider	0
extravagant_N	original, excentrique	1	fou	0
vol_2#I	acte criminel, crime, hold-up	1	combat	0
motte#I.1a	meule	1	ver	0
espoir#2	espérance	1	amitié	0
champ_1#II.2	parterre, étendue	1	millier	0
poisson#I.a	poiscaille	1	peu	0
yaourt#a	yogourt	1	dessert	0
vent#I.1	ouragan, cyclone, air, typhon	1	ciel	0
W.-C.#b	chiotte, waters, pipi-room, tartisses, commodités, goguenots, w.-c., chiottes, gogues, cabinet d'aisances, sanitaires, cabinets, latrines, chaise percée, tinettes, urinoir, lieux d'aisances, tinette, petit coin, wawas, toilettes, water-closet, cagoinces, petit endroit, vécés	1	toilette	0
assimilé#1	semblable	1	:	0
définir#II.1	caractériser, décrire	1	expliquer	1
japonais_N#II	nippon	1	français	0
menottes_1#I	lien	1	corde	0
cuisine#I.1	bouche	1	fête	0
maillet#1	marteau	1	ciseaux	0
rareté#II.2	objet, excentricité, pièce de collection	1	plante	0
criterium	course, tournoi	1	compétition	1
limite_N#II.2	seuil	1	heure	0
technologie	technique	1	physique	0
laitue#a	laitue	1	fraise	0
train_1#III.2	jeu, assortiment	1	paquet	1
courir#I.1a	se déplacer, voler, accourir, trotter, sprinter, cavaler, galoper, courir	1	marcher	1
couleur#IV.3	relief	1	sens	0
manger_V#I.2	détériorer, bouffer	1	envahir	0
rond_Adj#III	cuité, bourré, arraché, déchiré, ivre, casquette, pété, torché, blindé, saoul, plein, raide, beurré, pinté	1	seul	0
ici#I.a	près, partout, loin, ailleurs, quelque part, là-bas, par là, là, y	1	certain	0
tenir#VII.2	souhaiter, désirer, tenir à c9cur, vouloir	1	participer	0
pauvre_Adj#I	sans le sou	1	heureux	0
armoire#1	placard, armoire à glace, buffet, garde-robe, bahut, armoire normande, cave à vin	1	sac	0

judo	lutte, kung-fu, karaté, taekwondo, aïkido, jujitsu	1	natation	0
expédition#I	voyage	1	école	0
atermoyer	noyer le poisson, tergiverser, botter en touche	1	demander	0
chapelet#I.a	rosaire	1	rouleau	0
là#III.2	oh là là, allez !, oh !, ah !, eh !	1	paix	0
bouchon_1#IV	flotteur	1	barque	0
s'établir	s'installer	1	vivre	0
veste_2#1	déconfiture, échec, gamelle, défaite	1	retraite	0
coffre#I.1	caisse	1	lit	0
pachyderme#I.b	pachydermes, éléphant	1	singe	0
carré_N#I.1a	forme, carreau, case	1	morceau	0
brin#I	fil	1	poil	1
hypnotisme	hypnose	1	écriture	0
chaisière	chaisier	1	ouvrier	0
mépriser	cracher	1	connaître	0
graveleux	caillouteux	1	calcaire	0
équivalent_N	traduction, analogue	1	achat	0
goûter_N#I	quatre-heures, repas	1	rituel	0
cours_1#1a	leçon, conférence, classe	1	numéro	0
nuage#I	couverture, nues, nuée, mouton, brouillard, brume	1	vague	0
cou#I.1b	gaviot, gorge, encolure, gargamelle, kiki	1	bec	0
crado#V.2	sale, dégueu, dégueulasse, crasseux, crade	1	moi	0
apposer	fixer	1	graver	0
bouilloire	samovar	1	surprise	0
désert_N#X-II	vide, no man's land, friche, néant	1	pays	0
entassement#I.b	amoncellement, amas, tas, colline	1	accumulation	1
instaurer	définir, commencer	1	et	0
chaussette#II.1	filtre	1	machine	0
bol_1#I.a	écuelle	1	verre	0
augmentation#II	coup de chauffe, aggravation	1	écart	0
tourner#II	retourner	1	tenir	0
chiffons#II.b	habits, sapes, fringues	1	magasin	0
amante	maîtresse, amoureuse	1	lettre	0
tendance#II	vogue, courant, mode	1	vocation	0
adoratrice#II.1	fan	1	ami	0
pipe#III	turlute, fellation, pompier, gorge profonde, gâterie	1	moustache	0
marcher#II	avancer, progresser	1	manifester	0
limiter#1	borner	1	cacher	0
soucoupe#a	sous-tasse	1	le	0
pénétrer#I.1	entrer	1	souffler	0
se confirmer	se vérifier	1	reproduire	0
s'embarrasser	s'encombrer	1	manquer	0
galette#II.2	chapeau, couvre-chef	1	main	0
coup#V.2b	coup du père françois	1	ménage	0
enfant_Adj	jeune	1	seul	0
encadrer#III	border	1	et	0
siège#III.2	localisation	1	pouvoir	0
réussir#I.1	réaliser	1	conclure	0
cadre_1#I.1	encadrement	1	verre	0
analogue_Adj#a	près, connexe, parent, ressemblant, semblable, équivalent, analogue, voisin, similaire, comparable, proche	1	universitaire	0
dormir#I.2	coucher, pioncer	1	venir	0
bagouse	bague	1	plaque	0
casserole#II	casserolée	1	goutte	0
couché#III	sous la couette	1	debout	0
enjôleuse	séductrice	1	homme	0
inédit#I	inconnu	1	documentaire	0
champ_1#I.2	pré, champ	1	pâturage	1
là-dessus#II	là	1	soudain	0
bouffer_1#V.3	manger, avaler	1	être	0
nappe#III	séquence, série, suite	1	voix	0
semi-remorque	camion	1	moustique	0
se lever#IV.1b	arriver, commencer, débuter	1	produire	0
aspirateur_N#I.2	balayeuse, aspiratrice	1	trace	0
sale_Adj#I.2	cradoque, malpropre, crado, cracra, dégueu, dégueulasse, cradingue, crade	1	désordre	0
sentir#II	percevoir, pressentir	1	craindre	0
dans#III.1	pendant, durant	1	par	1
gymnastique_N#I.2	gym, séance	1	toilette	0
crado#IV	cradoque, sale, crado, cracra, dégueu, dégueulasse, salaud, cradingue, craspec, crade	1	bizarre	0
tiers_Adj	troisième	1	petit	0
ping-pong#II	table de ping-pong	1	immeuble	0
jumeau_N#III.2	macreuse, paleron, palette	1	terrine	0

libéré#I	libre	1	français	0
assis#III.2	stable, établi, fondé	1	connu	0
brosse#II.2	touffe	1	corbeille	0
dîner_N#II	souper, repas	1	plat	0
rapide_Adj#I.1	vif, leste, preste	1	supplémentaire	0
curieuse	questionneur, questionneuse	1	gens	0
trotte	trajet, déplacement, marche	1	pause	0
champagne#IV	beige	1	qui	0
figure#III.1a	personnage, triste sire, personnalité, figure de proue, chien fou	1	vision	0
déplacement#III.1	transport	1	oubli	0
cartouche#I	munitions, balle	1	lunette	0
laver#I.1	doucher, nettoyer, lessiver, savonner, rincer, shampouiner	1	relever	0
séducteur_Adj	enjôleur	1	vivant	0
vêtement#III.2	manteau	1	relais	0
boucler	attacher	1	prendre	0
encadrer#II.1a	entourer	1	noter	0
enfoncer#I.2	forcer, défoncer	1	fermer	0
musique#I.1b	zique, musique, zizique, air	1	recherche	0
laver#III.1	nettoyer	1	ouvrir	0
marche_1#III.1	fonctionnement	1	combustion	0
vélo#I	tandem, biclou, biclo, vélo tout terrain, bécane, bmx, vélo tout chemin, vtt, bicross, vélocipède, bicyclette, vtc, petite reine	1	dos	0
frère#II	père	1	saint	0
geyser#II.2	volcan, festival	1	tas	1
T.G.V.	train, train à grande vitesse	1	classique	0
s'endormir#III	décéder, tuer, périr, emporter, mourir, trépasser, disparaître, clamser, succomber, passer l'arme à gauche, perdre la vie, casser sa pipe, crever, s'éteindre, expirer	1	vivre	0
miraculeusement#2	par l'opération du saint-esprit	1	du	0
promesse#2	prédiction	1	certitude	0
félicité	contentement, bonheur, septième ciel	1	entente	0
sang-froid	calme	1	froid	0
au-dessous	au-dessus, en dessus, en dessous	1	près	0
discontinuité	intermittence	1	continuité	0
volcan#I	montagne de feu, guyot	1	centre	0
extase#1	septième ciel, bien-être, joie	1	harmonie	0
sincère#I	honnête, franc	1	public	0
pro-	pour, en faveur	1	nord	0
manger_V#I.2	détériorer, bouffer	1	prendre	0
séduisant#II	attrayant	1	magnifique	1
s'ennuyer#1	se barber, s'embêter, s'emmerder, se faire chier	1	jouer	0
curieusement#II	étrangement, bizarrement	1	très	0
butte#I	colline, pic, dune	1	forêt	0
par#II.1	à travers, via, en	1	sur	0
user#II	recourir	1	profiter	0
longueur#I.1	dimension	1	largeur	1
meeting#2	réunion, évènement sportif, rencontre	1	entraînement	0
arriver#II.2	atteindre	1	être	0
lupins#II	graine de lupin	1	tapas	0
pièce#I.1	rondelle, objet	1	forme	0
bouchon_1#III	ralentissement, embouteillage	1	carrefour	0
particulier_Adj#1	spécial, déterminé, spécifique	1	ultérieur	0
laitue#a	laitue	1	roquette	1
tuerie#II	catastrophe	1	erreur	0
savon_2#1	punition, volée de bois vert, raclée, volée	1	message	0
exercer#I.2	entraîner	1	aider	0
mouche#I	drosophile, mouche du vinaigre	1	fille	0
précis_Adj#2	exact	1	officiel	0
proche_Adj#III.1a	attaché, comme cul et chemise, intime	1	qui	0
servile	aplati, soumis	1	docile	1
symptôme#II	manifestation, indication, indice	1	signe	1
cité	ville	1	préfecture	0
bon-papa	bonne-maman, mémère, mère-grand, grand-mère, mémé, grand-papa, grand-maman, pépère, pépé, grands-parents, mamie, aïeule, grand-père, ancêtre, aïeul, papi	1	père	0
original_N#II	extravagant, drôle, excentrique, extraterrestre	1	comble	0
s'allonger_1#III	changer	1	exploser	0
conte#II	fable, salades, mensonge, histoire	1	miracle	0
cornichon_N#I	concombre	1	maison	0
prédateur_N	dévoreur	1	colonie	0
côté#II.1	flanc	1	pied	0
environnement#I	cadre, milieu	1	espace	1
lame#I	couperet, dent	1	version	0

sortir#II.1a	tirer	1	avoir	0
encouragement#1	soutien, incitation, exhortation, appui	1	joie	0
menottes_1#I	lien	1	porte	0
appétissant	alléchant	1	autre	0
transpirer	suer	1	marcher	0
animal_N#I.2	bête	1	chien	0
lisse_Adj#I	chauve, homogène	1	long	0
cogner#I.2	atteindre, frapper	1	taper	1
chèvre#IV	chevalet	1	pelle	0
excéder	dépasser	1	atteindre	0
testicule	couilles, bijoux de famille	1	cheveux	0
pré-#I	avant	1	pré	0
épier	espionner, observer	1	regarder	1
parler#II.2	causer	1	notre	0
respectable	bien-aimé, bien	1	sympathique	0
gargamelle	gaviot, gorge, kiki, cou	1	bouche	0
orage#II	agitation, tempête, nuages, trouble, perturbation, ouragan, tourmente	1	choc	0
voler_1#I.1	survoler, voleter, planer, voltiger, piquer, se déplacer	1	jouer	0
motocyclette	meule, mobylette, bécane, vélomoteur, cyclomoteur, engin, side-car, monture, pétoire, moto	1	voiture	0
sieste	sommeil	1	nuit	0
tartine#II.2	laïus	1	newsletter	0
voler_2#II	escroquer, flouer, rouler dans la farine, arnaquer, rouler, pigeonner, plumer, blouser, gruger	1	tromper	1
fluvial#b	portuaire, maritime	1	routier	0
idiot_Adj#I	bête, stupide, cornichon, sot	1	grand	0
manteau#I.a	capote, trois-quarts, blouson, tunique, veste	1	costume	0
fin_Adj#III.2	subtil	1	grossier	0
engager#III.1	encourager, déterminer	1	inviter	1
mesurer_1#II	évaluer, estimer	1	attribuer	0
voiture#2b	trajet, voyage, route, déplacement, autoroute	1	marche	0
se confronter	faire face	1	préparer	0
exagéré#II	limite	1	bizarre	0
potager	jardin	1	pays	0
lancer_V#II	jeter	1	pousser	0
abri#2	guitoune, refuge	1	usager	0
établi_N	banc	1	épaule	0
lin#II	graine de lin	1	palme	0
pègre	milieu	1	population	0
soupir#II	bruit	1	fouet	0
vagabonde	sans-logis, sans-abri	1	complètement	0
arrêter#V.2b	abandonner, jeter l'éponge	1	bien	0
canard_1#IV	biche	1	sac	0
privé_Adj	privatif, perso, personnel, intime	1	intérieur	0
truc#II	objet	1	torchon	0
facteur_1	coursier, messager, garçon de course, courrier, postier	1	transporteur	0
statuette	statue, figure	1	peinture	0
meeting#1	réunion publique, assises	1	des	0
pigeon#III	victime	1	prince	0
bouchon_1#V	chéri, chérie	1	bonhomme	0
coussin#1	oreiller	1	banc	0
téléphone#I	appareil téléphonique, bigophone	1	carnet	0
dûment	bien, ben	1	soigneusement	1
couleur#IV.3	relief	1	surprise	0
long_Adj#I.2a	interminable, tout en longueur, allongement, grand	1	noir	0

TABLEAU A.6 – Évaluation des candidats absents du DS 1 avec un score normalisé = 1 (méthode de base)

Entrée	**Lexs.**	**Score**	**Candidat**	**Syn**
âne#I	bourrique, baudet, bourriquet, bourricot	0.942099	chevaux	0
dormir#I.1a	en écraser, roupiller, pieuter, ronfler, pioncer	0.925219	suivre	0
dormir#I.1a	en écraser, roupiller, pieuter, ronfler, pioncer	0.876542	nuits	0
effrayer#2	craindre	0.960879	inquiéter	0
éléphant#I	pachyderme	0.947193	lion	0
encadrer#I.1a	encadrer	0.933979	passer	0
encadrer#I.1a	encadrer	0.945446	accrocher	0
enceinte_Adj	en cloque, grosse	0.968998	amoureux	0
escargot#I.a	limaçon, colimaçon	0.958362	insecte	0

ivre#I.1	casquette	0.944033	malade	0
ivre#I.1	casquette	0.939517	jeune	0
mignon_Adj	mimi, chou	0.967102	joli	1
narines#1	trou de nez	0.943524	na	0
odorat	nez, olfaction	0.960631	goût	0
patate#I.1	pomme de terre	0.925315	pomme	0
sans-abri#a	sans-logis	0.907360	mendiant	0
survêtement	jogging, survêt	0.878523	débardeur	0
thorax#a	cage thoracique	0.943458	corps	0
thorax#a	cage thoracique	0.933447	menton	0
W.-C.#a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.940880	wc	0
courir#I.2	foncer	0.862310	cour	0
adoptif#2	adoptif	0.855058	décédé	0
s'affoler	affoler	0.921620	crier	0
angoisser#1	angoisser, s'angoisser	0.930330	perturber	0
tatanes	grolles, pompes, chaussures, godasses	0.918443	escarpin	0
tram#I	tramway	0.943824	taxi	0
bécane#I.1	biclou, biclo, vélocipède, bicyclette, vélo, petite reine	0.910349	roue	0
autel	sainte table	0.930097	édifice	0
se désespérer	désespérer	0.916878	douter	0
temporal	temporo-	0.896107	latéral	0
pif	nez, nase, blair, tarin, naseaux, blase	0.876089	doigt	0
pif	nez, nase, blair, tarin, naseaux, blase	0.885723	truc	0
blair	pif, nez, nase, tarin, naseaux, blase	0.947513	regard	0
blase#II	pif, nez, nase, blair, tarin, naseaux	0.892831	sourire	0
arpion	ripatons, panards, pieds, pince	0.989743	pied	0
lolos	seins, nénés, tétons, nichons, roploplos, roberts, nibards, mamelles	0.942305	...	0
nénés	lolos, seins, tétons, nichons, roploplos, roberts, nibards, mamelles	0.995129	téton	0
nénés	lolos, seins, tétons, nichons, roploplos, roberts, nibards, mamelles	0.926201	ongle	0
nénés	lolos, seins, tétons, nichons, roploplos, roberts, nibards, mamelles	0.917025	pied	0
roberts	lolos, seins, nénés, tétons, nichons, roploplos, nibards, mamelles	0.902053	jumeau	0
roberts	lolos, seins, nénés, tétons, nichons, roploplos, nibards, mamelles	0.972882	croc	0
roberts	lolos, seins, nénés, tétons, nichons, roploplos, nibards, mamelles	0.934852	cheveux	0
fesse#III	cul	0.924784	f	0
océane	océanique	0.928068	océans	1
guibolles	cannes, pattes, gambettes, jambes	0.928060	tête	0
yogourt#a	yaourt	0.942770	fromage	0
s'égarer	se perdre	0.999007	aller	0
emprunteur_N	prêteur, emprunteuse, prêteuse	0.895605	emprunt	1
prêteur_N	emprunteur, emprunteuse, prêteuse	0.932114	prêt	1
souper_V	dîner	0.972621	faire	0
quatre-heures#I	goûter	0.880403	...	0
cabochard_Adj	têtu	0.969806	odieux	0
mastiquer	mâcher	0.966669	manger	0
résulter	causer, conduire, tenir, faire	0.857310	sortir	0
se fragiliser	fragiliser	0.971162	fracturer	0
gaspi	gaspillage	0.877439	gilet	0
furax	furieux	0.904763	fou	0
limaçon	escargot, colimaçon	0.854874	lézard	0
sculptage	sculpture	0.888373	dessin	0
matutinal	matinal	0.895279	blanc	0
rabibocher#II	raccommoder, réconcilier	0.851471	punir	0
raccommoder#II	rabibocher, réconcilier	0.894341	restaurer	0
tricherie	triche	0.888618	fraude	0
clocharde	clodo	0.938772	femme	0
dégueulasse#I.1	dégueu	0.917478	néfaste	1
guenilles	haillons	0.976438	noir	0
sale_Adj#V.1	salaud	0.874751	ça	0
lave-linge	machine	0.851548	...	0
bestiole#I	bête, bestiole	0.866323	créature	0
quéquette	zigounette, zizi, biroute, verge, teub, bite, pénis, membre, queue, membre viril, bistouquette, pine, braquemart, zob, manche, robinet, nouille, chibre, moineau	0.904501	main	0
bon-papa	bonne-maman, mémère, mère-grand, grand-mère, mémé, grand-papa, grand-maman, pépère, pépé, mamie, aïeule, grand-père, ancêtre, aïeul, papi	0.961021	papa	0
esgourdes	oreilles	0.927085	épaule	0
s'acheminer#I	se diriger	0.908424	marcher	0
arrestation	arrêt	0.859349	exécution	0
goguenots#a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.888550	plus	0

goguenots#a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.883998	pauvre	0
goguenots#a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.855739	rat	0
gogues#a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.970204	main	0
cagoinces#a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.961297	abeille	0
tartisses#b	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.930203	ruelle	0
wawas#a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.904892	moteur	0
mécanicien_N#I.2	mécano	0.887903	opérateur	0
T.G.V.	train à grande vitesse	0.891403	lycée	0
avalanche#III	ouragan	0.933857	tornade	1
gosier	pipe	0.918578	palais	0
Pays-Bas	hollande	0.865383	pays	0
tata#II	tapette, folle, tante, tantouze	0.903532	maman	0
tata#II	tapette, folle, tante, tantouze	0.889025	ta	0
miel_2	merde !	0.915704	moi	0
se dégarnir	se déplumer	0.895770	dé	0
bouffer_1#I.1b	manger	0.932842	faire	0
bouffer_1#III.1	se bouffer, manger	0.859968	mordre	0
bouffer_1#VI.4	manger	0.993616	manquer	0
bouffer_1#IV.1	se bouffer, manger	0.862161	dévorer	0
F6	t6, six-pièces	0.935411	appartement	0
F6	t6, six-pièces	0.871537	appart	0
T2	f2, deux-pièces	0.947009	studio	0
T5	cinq-pièces, f5	0.883593	t	0
T10	f10, dix-pièces	0.963680	studio	0
raccourcir#II	réduire, diminuer	0.894446	barrer	0
café-concert	caf'conc'	0.937380	cabaret	1
amarante#II	pourpre	0.932574	noire	0
crasseux#II	cradoque, crado, cracra, cradingue, craspec, crade	0.937576	mauvais	0
réprimander	botter les fesses	0.916863	punir	0
cradingue#I.2	crado, crade, cracra, cradoque	0.856010	dégueulasse	0
cradingue#IV.2	cradoque, crado, cracra, crasseux, cradingue, craspec, crade	0.935481	excentrique	0
cradingue#IV.2	cradoque, crado, cracra, crasseux, cradingue, craspec, crade	0.889933	vulgaire	0
cradingue#IV.2	cradoque, crado, cracra, crasseux, cradingue, craspec, crade	0.968169	gothique	0
cradingue#IV.2	cradoque, crado, cracra, crasseux, cradingue, craspec, crade	0.928123	déchiré	0
craspec#V.1	crade	0.974125	dégueulasse	0
craspec#V.1	crade	0.922378	gras	0
craspec#V.1	crade	0.890597	spectaculaire	0
craspec#V.2	cradoque, crado, cracra, crasseux, cradingue, craspec, crade	0.972797	pec	0
craspec#V.3	cradoque, crado, cracra, crasseux, cradingue, crade	0.880170	sympa	0
craspec#V.3	cradoque, crado, cracra, crasseux, cradingue, crade	0.864672	lourd	0
crade#I.2	cradingue, crado, cracra, cradoque	0.976652	vulgaire	0
crade#I.2	cradingue, crado, cracra, cradoque	0.853511	glauque	0
crade#V.2	craspec	0.882263	ridicule	0
crade#VI.1	cradoque, crado, cracra, crasseux, cradingue, craspec, crade	0.909921	sale	1
crade#VI.2	crado, crasseux	0.977012	moche	0
crade#VI.2	crado, crasseux	0.905272	dégueulasse	0
cradoque#IV	crado, cracra, cradingue, craspec, crade	0.994423	rouge	0
cradoque#IV	crado, cracra, cradingue, craspec, crade	0.943237	cra	0
dégueu#VI.3	dégueulasse	0.912171	chiant	0
dégueu#VI.1	dégueulasse	0.899800	ridicule	0
dégueu#VI.4	dégueulasse	0.960888	drôle	0
dégueu#VI.2	dégueulasse	0.945569	ridicule	0
dégueu#V.1	dégueulasse	0.983695	drôle	0
dégueu#V.1	dégueulasse	0.944913	d	0
dégueu#V.1	dégueulasse	0.895258	...	0
dégueu#V.1	dégueulasse	0.961264	horrible	1
dégueu#V.1	dégueulasse	0.999721	beau	0
dégueu#V.1	dégueulasse	0.961102	moche	1
dégueu#V.2	dégueulasse	0.863454	mauvais	1
dégueu#II.2	dégueulasse	0.904346	</s>notused	0
dégueu#II.2	dégueulasse	0.885218	bien	0
États-Unis	amérique, usa	0.930638	uni	0
États-Unis	amérique, usa	0.889872	de	0
deudeuche	deuche, deux-chevaux, 2 cv	0.864922	femme	0

W.-C.#b	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.985764	toilette	0
water-closet#b	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.859334	wc	0
goguenots#b	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.896325	bois	0
gogues#b	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.892257	escalier	0
gogues#b	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.876680	pied	0
pâtisson	bonnet de prêtre	0.982995	courgette	0
tartisses#a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.884085	clé	0
fèces	excréments, selle, bol fécal, matière fécale	0.928390	sang	0
fèces	excréments, selle, bol fécal, matière fécale	0.868602	sperme	0
pioncer#II	dormir, roupiller, somnoler	0.880264	étudier	0
paresser	glandouiller, glander	0.854528	chauffer	0
ambages	détour	0.873442	accent	0
macchab	cadavre, macchabée, dépouille	0.965942	homme	0
rocking-chair	chaise à bascule, berceuse, chaise berçante, fauteuil à bascule	0.968234	rock	0
bosseur	travailleur	0.859717	boss	0
amollir	ramollir	0.947180	dissoudre	0
rhinocéros	rhino	0.878159	oiseau	0
-phobie#1	phobie	0.851944	-	0
CNCC	compagnie nationale des commissaires aux comptes	0.948608	cci	0
GIE	groupement d'intérêt économique	0.910072	entreprise	0
GIE	groupement d'intérêt économique	0.907615	groupe	0
hollandais_Adj#I.2	néerlandais	0.893881	belge	0
marital#b	conjugal	0.945298	matrimonial	1
se chipoter	se quereller, s'engueuler, se disputer	0.963522	trouver	0
coccinelle#I	bête à bon dieu	0.871315	papillon	0
toasteur	grille-pain	0.851742	trépied	0
magnet	aimantin	0.875277	réfrigérateur	0
magnet	aimantin	0.866925	aimant	1
amphibiens	batraciens	0.999525	homme	0
amphibiens	batraciens	0.945592	</s>notused	0
macaroni#II.a	italien	0.900362	con	0
encadrer#IV.a	encadrer	0.939519	cadrer	0
enjôler#a	enjôler	0.857958	insulter	0
enjôler#b	enjôler	0.871480	flatter	0
crico-	cricoïdien	0.991724	intra	0
encadrer#IV.b	encadrer	0.934765	définir	0
réactant	réactif	0.966200	carbone	0
gamberger	réfléchir	0.890900	sourire	0
gamberger	réfléchir	0.857348	marcher	0
liaison#II	liaison chimique	0.978767	élément	0
creusé#a	creusé	0.887287	développé	0
caca_N#III	pétrin, merde	0.870784	respect	0
alysse	corbeille d'argent	0.991967	...	0
alysse	corbeille d'argent	0.924234	rose	0
fuchsia#II.a	rose fuchsia	0.868152	rose	0
fuchsia#II.a	rose fuchsia	0.892613	rose	0
gaillet	caille-lait	0.947252	pin	0
gaillet	caille-lait	0.869280	papillon	0
cailler#I.1	se cailler	0.884892	lait	0
cailler#I.2a	cailler	0.920712	casser	0
cailler#I.2a	cailler	0.877763	couler	0
cailler#II	se cailler, geler, se geler les fesses	0.909454	ca	0
se cailler#II	geler, se geler les fesses, cailler	0.967285	crever	0
caille-lait	gaillet	0.946607	basilic	0
caille-lait	gaillet	0.859225	romarin	0
caille-lait	gaillet	0.978182	autres	0
lupins#II	graine de lupin	0.982655	courgette	0
patronyme#2	nom	0.883927	surnom	0
collabo_N#b	collaboratrice	0.923463	fasciste	0
valoches#II	valises	0.986066	boucle	0
valoches#II	valises	0.879799	patte	0
valoches#II	valises	0.856995	puce	0

TABLEAU A.7 – Évaluation des candidats absents du DS 0 avec un score normalisé ≥ 0,85 < 1 (méthode <SEP>)

Entrée	Lexs.	Score	Candidat	Qsyn
F6	t6, six-pièces	0.871537	appart	1
tracasser#2	chipoter, inquiéter	0.997975	déranger	0
privatiser	dénationaliser	0.946927	privatisation	1
nager#II.2	baigner, se trouver, croupir, accueillir, tremper	0.996523	marcher	0
se solder	finir	0.952007	porter	0
avaler#V	bouffer, se bouffer, manger	0.920158	valoir	0
sans-emploi#b	chômeuse	0.914309	sans	0
tatanes	pompes, sabot, souliers, chaussons, grolles, savates, chaussures, godasses	0.918443	escarpin	1
aérosol	bombe	0.915400	peinture	0
café-concert	caf'conc'	0.937380	cabaret	1
océane	océanique	0.928068	océans	1
merveilleusement	formidablement	0.995072	très	0
honteusement#II	scandaleusement	0.894439	trop	0
ostentatoire	ostensible, voyant	0.905539	excessif	0
âne#I	bourrique, baudet, bourriquet, bourricot	0.942099	chevaux	0
gaillet	caille-lait	0.947252	pin	0
ébahissant	surprenant, étonnant	0.863481	grandiose	0
F6	t6, six-pièces	0.935411	appartement	1
colorier	colorer	0.966674	faire	0
boitement	claudication	0.990964	saut	0
fèces	matière fécale, crotte, bouse, étron, selle, merde, caca, cigare, excréments, colombin, bol fécal	0.928390	sang	0
imprimer#I	marquer	0.939881	avoir	0
ping-pong#I.1	tennis de table	0.883938	billard	0
achever#II	parachever, terminer, finir	0.851288	continuer	0
s'étendre#II	s'arrêter, aller, avancer, s'allonger, arriver, s'engager, courir, s'étaler	0.867174	s	0
arrestation	arrêt	0.859349	exécution	0
s'acheminer#I	se diriger, piquer du nez, piquer	0.908424	marcher	0
mésange	mésange boréale, mésange huppée, mésange nonnette, mésange charbonnière, mésange bleue	0.861884	oiseau	1
voyou_N#2	garnement, vaurien, filou, crapule, bandit, chenapan, canaille	0.879351	voleur	0
vanter	louer	0.953657	rappeler	0
arpion	ripatons, panards, pieds, pince	0.989743	pied	0
musicienne	joueuse, interprète, virtuose	0.896730	musicien	0
assombrir#I	obscurcir	0.969856	dominer	0
dégueu#VI.3	cradoque, sale, crado, dégueulasse, crasseux, crade	0.912171	chiant	0
souper_V	bouffer, se bouffer, manger, dîner, se manger	0.972621	faire	0
se chipoter	se quereller, s'engueuler, se disputer	0.963522	trouver	0
déhanchement	coup de reins	0.893725	numéro	0
chèvre#III	grue	0.941896	ossature	0
nager#I.c	nager	0.957717	courir	0
résulter	blesser, coucher, exciter, susciter, entraîner, inspirer, impliquer, conduire, tenir, apporter, dépendre, maintenir, causer, faire, se coucher, remplir, provoquer, déclencher, pousser, se déclencher	0.857310	sortir	0
essaim	nuage, colonie	0.917940	abeille	0
chipoteur_Adj#II.2	exigeant	0.868239	chip	0
inconvénient#I	blème, problème	0.879648	avantage	0
fumeux	fumant	0.861130	fum	0
élucider	résoudre	0.876524	explorer	0
encadrer#V	entourer	0.944026	entrer	0
receveur	bac	0.955695	paroi	0
satrape	tyran	0.995884	esclave	0
analogie#c	similitude, ressemblance	0.989254	comparaison	1
terrorisme#II	domination	0.860257	signe	0
guitoune#2	abri, cabane	0.900975	loge	1
grimper#I.3	prendre place	0.952550	monter	1
manteau#I.a	capote, trois-quarts, blouson, tunique, veste	0.950358	vêtement	1
neigeux#2	enneigé	0.932994	neige	1
circuler#I.1	rouler	0.989045	lait	0
gonzesse	meuf, femme, femelle	0.980626	salope	0
bien_N#IV.b	bien	0.890651	actif	0
malpropre#II.2	sale	0.934310	dangereux	0
puy	colline	0.856022	mouton	0
ivre#I.1	cuité, rond, bourré, arraché, déchiré, casquette, pété, blindé, saoul, torché, plein, raide, entre deux vins, beurré, pinté	0.944033	malade	0
rôder#II	menacer, planer	0.864829	régner	0

dégueu#V.1	dégueulasse	0.944913	d	0
goguenots#a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, latrines, chaise percée, tinettes, urinoir, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.855739	rat	0
vermine#II	insecte	0.861350	pute	0
insuffisamment#II	mal	0.944075	cruellement	0
anarchie#II	désordre, souk, chaos	0.875379	ordre	0
valoches#II	valises, poches, cernes	0.856995	puce	0
écarter#III.3	exclure	0.960609	sortir	0
affûter	aiguiser	0.943603	prendre	0
parfumeuse	nez	0.904971	styliste	0
lointain_Adj#II	éloigné	0.859032	possible	0
valet#II	carte, figure	0.927582	chevalier	0
injure#1	insulte	0.874397	«	0
mangeable	consommable, immangeable	0.852967	manger	1
scier#III.1	blesser	0.889776	briser	1
rigolard_N	rigoleur	0.881083	rire	0
acerbe	à l'emporte-pièce	0.920443	violent	0
cailler#I.1	se cailler, se dégrader	0.884892	lait	0
stéatopyge	fessu	0.973935	obèse	0
orphelin_Adj#I.1b	orphelin, abandonné	0.943759	humain	0
gnognote	petite bière	0.997313		0
planer_1#I.2	flotter	0.982396	être	0
gonfler#II	se gonfler, lever	0.870145	tenir	0
fesses#I.a	siège, cul, paire de fesses, séant, postérieur, derrière, popotin, derche, lune, miches, pétard, fessier, croupion, arrière-train, croupe, ballons, panier	0.897818	cuisse	0
répandre#I.2	provoquer	0.965473	faire	0
infarctus	crise cardiaque	0.932575	</s>notused	0
lessive#III	linge	0.879699	laine	0
calembour	blague, jeu de mots	0.879263	commentaire	0
gravitation	attraction	0.910024	gravité	0
esgourdes	oreilles	0.927085	épaule	0
cornet#a	barquette	0.948129	chaudron	0
perpétuellement	éternellement	0.951860	constamment	1
excentricité_1#II.1	vogue, nouveauté, mode	0.934636	sobriété	0
s'entretenir_1	causer, discuter, converser	0.923798	parler	1
analogie#c	similitude, ressemblance	0.889753	elle	0
entêté_Adj	cabochard, têtu	0.997480	désespéré	0
États-Unis	amérique, usa	0.889872	de	0
ligoter	menotter, lier	0.942548	capturer	0
fabulatrice	fabulateur	0.855079	prostitué	0
éléphant#II.2	ponte, ténor	0.898938	tigre	0
solarisation	insolation	0.850141	coloration	0
carder	peigner	0.917236	de	0
contenu_Adj	étouffé, retenu	0.905076	muet	0
amphibiens	batraciens	0.999525	homme	0
raccourcir#II	réduire, diminuer	0.894446	barrer	0
marital#b	conjugal	0.945298	matrimonial	1
frugal#1	sur le pouce	0.882906	gourmand	0
tertre	colline	0.922654	bassin	0
inconvénient#I	blème, problème	0.946664		0
sire#II	seigneur, individu	0.982338	homme	1
s'étouffer	s'étrangler	0.955117	dormir	0
trahir	poignarder dans le dos, passer à l'ennemi	0.888637	ruiner	0
rabibocher#II	raccommoder, réconcilier	0.851471	punir	0
vocifération	rugissement, grognement, grommellement	0.951499	voix	0
dodu#2	rebondi	0.879053	...	0
bien-aimé_Adj#I.2	respectable	0.864720	aimé	0
volte-face#II	changement	0.975868	mensonge	0
gamberger	réfléchir, penser	0.857348	marcher	0
rocking-chair	chaise à bascule, berceuse, chaise berçante, fauteuil à bascule	0.968234	rock	0
systématiquement#I	méthodiquement	0.988350	successivement	0
se détendre#I	se relâcher, se décontracter	0.998373	tendre	0
orphelin_Adj#I.1b	orphelin, abandonné	0.931367	perdu	0
prête-nom	mandataire	0.896228	surnom	0
shampouiner#1	laver	0.854060	reprendre	0
s'engager#II.2	démarrer, commencer, s'ouvrir, débuter	0.975123	engager	0
aliter#2	clouer au lit	0.929058		0
paresser	se tourner les pouces, glandouiller, roupiller, somnoler, dormir, glander, pioncer, buller	0.854528	chauffer	0
T5	cinq-pièces, f5	0.883593	t	0
cailler#I.2a	cailler	0.920712	casser	0

nappe#III	séquence, série, suite	0.897060	voix	0
remunérer	payer	0.889339	percevoir	1
latrines#a	chiotte, waters, tartisses, pipi-room, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.889259	douche	0
deudeuche	deuche, deux-chevaux, 2 cv	0.864922	femme	0
verbeux	prolixe, fleuve	0.919886	ennuyeux	0
dent_1#II.4	aiguille, sommet, piton, pic	0.910813	de	0
observer#III	respecter	0.974163	vivre	0
embarrasser#I	encombrer	0.952675	attendre	0
honnête#II.a	correct, comme il faut	0.947191	digne	0
laitue#b	laitue	0.913788	roquette	1
jeter#V.1	lancer	0.904233	faire	0
folichon#I	gai	0.890921	fol	0
frissonnant#I	grelottant	0.887946	seul	0
s'établir	s'installer	0.915578	résider	0
petit-fils	arrière-petit-fils	0.968513	enfant	0
héliporter	hélitreuiller	0.871899	</s>notused	0
s'engager#I.1	entrer, rejoindre	0.864675	rester	0
cogner#V.2	battre	0.970232	c	0
myocarde	muscle	0.993482	coeur	1
extorsion	chantage	0.899613	échange	0
cradoque#VI.1	dégueulasse, dégueu, sale	0.922379	cool	0
trembleur	tremblant	0.977007	nu	0
menottes_1#I	lien	0.931930	botte	0
ostentatoire	ostensible, voyant	0.930320	excessif	0
s'allonger_1#II.2	rallonger	0.866176	croître	1
tata#II	homosexuel, tantouze, pédale, pédé, lopette, lope, homo, folle, tapette, tante, gay	0.903532	maman	0
entêté_Adj	cabochard, têtu	0.961486	épuisé	0
ébrieux	alcoolique	0.860696	effrayant	0
se désertifier#I	se transformer	0.905688	disparaître	0
lessive#II.1	nettoyant, détergent	0.860575	gamme	0
réverbération#II	écho	0.887940	diffusion	0
toundra	désert, plaine	0.885318	...	0
soulever#I.2	lever	0.895523	soulev	0
dégueu#V.1	dégueulasse	0.983695	drôle	0
clocharde	sans-logis, sdf, clodo, sans-abri, sans domicile fixe	0.938772	femme	1
couché#V	penché	0.953724	écrit	0
inonder#II	baigner	0.953832	submerger	1
flotter#I	nager, surnager	0.912260	couler	0
pif	trou de nez, narines, naseau, nez, nase, blair, museau, tarin, naseaux, blase	0.885723	truc	0
embrouiller	noyer le poisson	0.901993	tromper	1
guignol#I.2	marionnette	0.930619	personnage	0
mastiquer	chiquer, mâcher	0.966669	manger	0
décolorer#I.1b	colorer	0.857116	laver	0
tricherie	triche	0.888618	fraude	1
cradingue#IV.2	cradoque, sale, crado, cracra, dégueu, dégueulasse, crasseux, cradingue, craspec, crade	0.935481	excentrique	0
boucler	attacher	0.893124	serrer	0
dru#I	dense	0.995172	plein	0
s'embêter	s'ennuyer, se barber, se faire chier, s'emmerder	0.977999	faire	0
épousée	épouse	0.968500	enceinte	0
vestibule	entrée, hall	0.910447	plafond	0
bouchon_1#I.1a	tampon	0.851464	bouteille	0
nautisme	bateau	0.937634	n	0
entêté_Adj	cabochard, têtu	0.942810	déçu	0
déprimant	cafardeux	0.998892	ennuyeux	0
fuselage	carlingue	0.904339	cockpit	1
lune#I.1a	lune	0.907518	l	0
détrousser	voler	0.916808	envahir	0
laitue#a	laitue	0.933950	salade	1
accessoirement	en outre	0.935530	indirectement	0
cravate#III	coup, prise	0.903004	casque	0
bouffer_1#I.1b	casser la graine, s'alimenter, grailler, casser la croûte, manger, tortorer, becter, se nourrir, se sustenter	0.932842	faire	0
goguenots#b	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, latrines, chaise percée, tinettes, urinoir, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	0.896325	bois	0
hélistation	héliport, aéroport	0.945855	plateforme	0
hanter#III	déchirer, perturber, tourmenter, angoisser, s'angoisser	0.896157	être	0
asservissement	esclavage	0.996600	détention	0
amante	maîtresse, amoureuse	0.888710	amant	1
rugueux	rude	0.853762	tranchant	0
accoupler#I	réunir	0.866979	combiner	1

nettoyer#I.4	laver	0.889897	secouer	0
racloir	grattoir	0.997129	aspirateur	0
couperet	lame	0.884398	levier	0
étouffement#I	tasse	0.852482	infection	0
calembour	blague, jeu de mots	0.965506	connerie	0
surdimensionné	grand	0.987382	...	0
survêtement	jogging, survêt	0.878523	débardeur	0
repérable	visible	0.930697	identifiable	1
menotte_1#II	attache	0.895801	chaîne	0
paroxysme	summum	0.960338	cœur	0
marcher#I.2	défiler	0.892653	aller	1
braire#I	crier	0.998845	se	0
vanter	louer	0.864019	dire	0
lessive#III	linge	0.988536	lave	0
galoper#II	trotter, courir	0.939214	galop	1
chiffrage	inventaire	0.939400	bilan	1
réverbération#II	écho	0.883524	restitution	0
encadrer#I.1a	encadrer	0.933979	passer	0
jeter#V.1	lancer	0.899321	mettre	0
nénés	lolos, seins, poitrine, sein gauche, tétons, nichons, roploplos, roberts, sein droit, nibards, mamelles	0.917025	pied	0
se cailler#II	avoir froid, se geler les fesses, geler, cailler	0.967285	crever	0
immobilisations#II	bien	0.923874	équipement	0
compatissant	compréhensif, empathique	0.940237	compétent	0
allonger_1#I.1	élargir, étendre, rallonger, étirer, agrandir	0.980077	prolonger	0
soupçonneux	méfiant	0.998637	fou	0
se nourrir#2	bouffer, s'alimenter, manger	0.902856	ssez	0
s'enfoncer#II	plonger, sombrer, tomber	0.952677	rentrer	1
alysse	corbeille d'argent	0.924234	rose	0
despote	tyran	0.994140	roi	1
bien-aimé_Adj#I.2	respectable	0.919279	bien	0
braire#I	crier	0.973933	manger	0
orpheline#b	abandonnée, orpheline	0.914179	enfant	0
sensibilité#IV.3	fibre, conviction, idéologie	0.940753	personnalité	0
-phobie#1	peur, terreur, phobie, angoisse	0.851944	-	0
factrice_1	postière, messagère, coursière	0.906438	vendeuse	0
baragouiner#I.2	parler	0.853790	apprendre	0
s'étendre#II	s'arrêter, aller, avancer, s'allonger, arriver, s'engager, courir, s'étaler	0.851580	étendre	0
accessoirement	en outre	0.942915	ensuite	0
kérosène	carburéacteur, combustible, carburant	0.976381	diesel	1
perturbant#1	gênant	0.901984	furieux	0
chandail	pull, pull-over	0.966969	robe	0
remontrance	savon	0.970861	réaction	0
gober#I	bouffer, se bouffer, se manger, manger	0.964567	prendre	0
fesse#III	cul, sexualité	0.924784	f	0
angoissé_Adj#2	anxieux	0.984235	impatient	0
inspirateur_Adj	inspirant	0.892506	créateur	0
thorax#a	poitrine, cage thoracique, coffre, buste, torse	0.933447	menton	0
marcassite	pierre de tonnerre	0.854130	quartz	0
caresser#IV	nourrir	0.856741	réaliser	0
ivre#I.2	rempli	0.874381	amoureux	0
canaille#1	escroc, filou, crapule, brigand, bandit, gangster, misérable	0.959031	chienne	0
coin-cuisine	cuisine	0.894567	four	0
endormir#b	assommer	0.926486	...	0

TABLEAU A.8 – Évaluation des candidats absents du DS 1 avec un score normalisé ≥ 0,85 < 1 (méthode <SEP>)

Entrée	**Lexs.**	**Score**	**Candidat**	**Syn**
caf'conc'	café-concert	1	café	0
grand-père#I	bon-papa, bonne-maman, mémère, mère-grand, grand-mère, mémé, grand-papa, grand-maman, pépère, pépé, mamie, aïeule, ancêtre, aïeul, papi	1	père	0
se chipoter	se quereller, s'engueuler, se disputer	1	piquer	0
louage	location	1	nuit	0
dégueu#V.1	dégueulasse	1	mignon	0
exposer#II	présenter	1	expliquer	1
car_N	autocar	1	bus	1
bécane#I.2	motocyclette, meule, moto	1	voiture	0

cadette	puînée	1	cadet	0
encadrer#I.1b	encadrer	1	entourer	1
aspirer_1#I.1b	inspirer, aspirer	1	respirer	0
dégueu#V.1	dégueulasse	1	bizarre	0
bière_1#b	binouze	1	bonne	0
pépère#II	pépé, grand-père, papi	1	con	0
fiacre	sapin	1	bus	0
inaudible	non audible	1	irrésistible	0
cradingue#VII	cradoque, crado, cracra, crasseux, craspec, crade	1	déjanté	0
furibond	furibard	1	débordé	0
grand-père#II	pépère, pépé, papi	1	père	0
vouvoyer	voussoyer	1	reconnaître	0
dégueu#VI.3	dégueulasse	1	honteux	0
qqch	quelque chose	1	quoi	0
actinidia	kiwi	1	arbre	0
parterre#II.2	champ	1	buisson	0
s'éveiller	se réveiller	1	revenir	0
payse#II	fiancée, promise	1	patrie	0
équitation	cheval	1	équestre	1
chibre	zigounette, zizi, biroute, verge, teub, bite, pénis, membre, queue, membre viril, quéquette, bistouquette, pine, braquemart, zob, manche, robinet, nouille, moineau	1	clitoris	0
pif	nez, nase, blair, tarin, naseaux, blase	1	pet	0
moche_Adj	laid	1	beau	0
commémorer#2	commémorer	1	célébrer	0
amollir	ramollir	1	fondre	0
cinoche#III	cinéma, comédie	1	signe	0
conteuse#1	raconteuse	1	conteur	0
éléphant#I	pachyderme	1	chevaux	0
colérique_Adj	coléreux	1	méchant	0
coccinelle#I	bête à bon dieu	1	abeille	0
uriner	pisser	1	être	0
s'appeler#II	être	1	signifier	1
jam-session	bœuf, jam	1	j	0
cracra#III	cradoque, crado, crasseux, cradingue, crade	1	pâle	0
encadrer#IV.b	encadrer	1	pour	0
escargot#I.a	limaçon, colimaçon	1	fourmi	0
ouest_N#I	occident, ponant	1	est	0
mignon_Adj	mimi, chou	1	mignonne	0
cheval#I.1a	dada	1	chevaux	0
crade#II	cradoque, crado, cracra, crasseux, cradingue, craspec, crade	1	con	0
menotte_2	mimine	1	main	1
F9	neuf-pièces, t9	1	f	0
sans-abri#b	sans-logis	1	femme	0
manger_V#I.1a	bouffer, se bouffer, se manger	1	boire	0
vécés#b	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	chat	0
wawas#a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	w	0
se cailler#II	geler, se geler les fesses, cailler	1	mordre	0
tocante	montre	1	to	0
cracra#I.2	cradingue, crado, crade, cradoque	1	cra	0
limaçon	escargot, colimaçon	1	loup	0
réactant	réactif	1	composant	0
dégueu#VI.4	dégueulasse	1	ridicule	0
F3	trois-pièces, t3	1	f	0
dégueu#III	dégueulasse	1	...	0
tante#I	tata, tatie, tantine	1	mère	0
creusé#a	creusé	1	profond	0
passerose	rose trémière	1	rose	0
manger_V#I.1a	bouffer, se bouffer, se manger	1	être	0
ci-dessus	susvisé	1	suivantes	0
tante#II	tapette, folle, tata, tantouze	1	mère	0
gamberger	réfléchir	1	nager	0
toasteur	grille-pain	1	four	0
dégueu#V.2	dégueulasse	1	moche	1
wawas#b	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	meuble	0
s'appeler#I.2	se nommer	1	être	0
mécanicienne#I.1	machiniste	1	mécanicien	0
tante#I	tata, tatie, tantine	1	mère	0
huit-pièces	t8, f8	1	appartement	0

colibri	oiseau-mouche	1	oiseau	0
se chipoter	se quereller, s'engueuler, se disputer	1	...	0
colvert	malard	1	oiseau	0
lave-linge	machine	1	lave	0
gaspi	gaspillage	1	bazar	0
puer#I	schlinguer	1	sentir	0
éléphant#I	pachyderme	1	lion	0
avalanche#III	ouragan	1	multitude	1
réactif_N#1	réactant	1	composé	0
encadrer#I.1a	encadrer	1	peindre	0
popotin	cul, séant, paire de fesses, postérieur, derrière, lune, derche, miches, pétard, fessier, croupion, arrière-train, croupe, panier	1	ventre	0
tinettes#b	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	voiture	0
BOFiP	bulletin officiel des finances publiques	1	bo	0
bouffer_1#V.3	manger	1	dévorer	0
mécanicien_N#II.1	mécano	1	technicien	0
réactant	réactif	1	composant	0
sans-abri#a	sans-logis	1	réfugié	0
cracra#I.1	cradoque, crado, crasseux, cradingue, craspec, crade	1	cra	0
crade#I.2	cradingue, crado, cracra, cradoque	1	ridicule	0
picrate	pinard, vin	1	pic	0
odorat	nez, olfaction	1	palais	0
W.-C.#b	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	wc	0
bouffer_1#III.1	se bouffer, manger	1	étouffer	0
purgatoire#II	désert	1	calvaire	1
venger#I	venger	1	ven	0
dépotoir#I	décharge	1	atelier	0
deuche	deudeuche, deux-chevaux, 2 cv	1	bagnole	0
incurable	inguérissable	1	mental	0
lave-pont	balai-brosse	1	balai	0
lune#III	cul, séant, paire de fesses, postérieur, derrière, popotin, derche, miches, pétard, fessier, croupion, arrière-train, croupe, panier	1	l	0
babines#2	lèvres	1	lèvre	0
crado#IV	cradoque, cracra, cradingue, craspec, crade	1	cra	0
réactant	réactif	1	glucose	0
rhinocéros	rhino	1	chevaux	0
blatte	cafard	1	rat	0
profonde	poche	1	tiroir	0
se chipoter	se quereller, s'engueuler, se disputer	1	regarder	0
thorax#a	cage thoracique	1	ventre	0
CMPU	coût moyen unitaire pondéré	1	stock	0
batraciens#a	amphibiens	1	insecte	0
fiancée	payse, promise	1	fiancé	0
prêteuse	emprunteur, prêteur, emprunteuse	1	prête	0
costard-cravate	costume-cravate	1	costume	1
déguster#II	savourer	1	recevoir	0
se vêtir	se saper, s'habiller, se fringuer	1	couvrir	0
transatlantique	transat, chaise longue	1	...	0
quenotte	ratiche, dent	1	chaussette	0
encadrer#IV.a	encadrer	1	séparer	0
fuchsia#II.a	rose fuchsia	1	blanc	0
grand-père#I	bon-papa, bonne-maman, mémère, mère-grand, grand-mère, mémé, grand-papa, grand-maman, pépère, pépé, mamie, aïeule, ancêtre, aïeul, papi	1	père	0
dégueu#VI.4	dégueulasse	1	ridicule	0
s'endormir#III	décéder, mourir, emporter, périr, trépasser, disparaître, clamser, expirer, perdre la vie, casser sa pipe, crever, tuer	1	dormir	0
enjôler#b	enjôler	1	intriguer	0
s'angoisser	angoisser	1	paniquer	0
batraciens#a	amphibiens	1	crustacé	0
naseaux#II	pif, nez, nase, blair, tarin, blase	1	narine	1
étreindre#I.1	embrasser	1	arracher	0
doctoresse	médecin, docteure, doc, femme médecin, toubib	1	infirmière	0
carrousel	chevaux de bois	1	manège	0
s'égarer	se perdre	1	entrer	0
génitrice	procréatrice	1	mère	1
tante#I	tata, tatie, tantine	1	mère	0
nibards	lolos, seins, nénés, tétons, nichons, roploplos, roberts, mamelles	1	...	0
taillage	taille	1	perçage	0
magnet	aimantin	1	frigo	0
dormir#I.2	pioncer	1	vivre	0
ivre#I.1	casquette	1	alcoolique	0
éléphant#I	pachyderme	1	chat	0

W.-C.#b	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	wc	0
PEPS	premier entré, premier sorti	1	pe	0
vioque	vieille	1	vache	0
W.-C.#a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	wc	0
pj_1	pièce jointe	1	p	0
ping-pong#II	table de ping-pong	1	billard	0
humain_N	homme	1	humains	0
autobus	bus	1	autocar	1
trois-pièces#III	f3, t3	1	studio	0
déshydraté#2	lyophilisé	1	sauvage	0
Nippone	japonaise	1	française	0
mécanicienne#I.2	mécano	1	mécanicien	0
alimenter#II	marcher	1	aliment	0
eau-de-vie	gnôle	1	eau	0
OPA	offre publique d'achat	1	op	0
escargot#I.a	limaçon, colimaçon	1	taureau	0
boire_V#I.2b	picoler	1	consommer	1
bifteck#I	steak	1	poulet	0
cracra#III	cradoque, crado, crasseux, cradingue, crade	1	orangé	0
dégueu#VI.2	dégueulasse	1	grotesque	0
sale_Adj#V.1	salaud	1	cela	0
malfaiteur	malfrat	1	personne	0
mémère#I	bon-papa, bonne-maman, mère-grand, grand-mère, mémé, grand-papa, grand-maman, pépère, pépé, mamie, aïeule, grand-père, ancêtre, aïeul, papi	1	mère	0
clocharde	clodo	1	prostitué	0
blair	pif, nez, nase, tarin, naseaux, blase	1	sang	0
amphibiens	batraciens	1	animal	0
s'allonger_2#I.b	se coucher	1	courir	0
encadrer#I.1b	encadrer	1	cadrer	0
ouragan#II.2	tornade	1	cyclone	1
grand-mère#I	bon-papa, bonne-maman, mémère, mère-grand, mémé, grand-papa, grand-maman, pépère, pépé, mamie, aïeule, ancêtre, grand-père, aïeul, papi	1	mère	0
W.-C.#b	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	w	0
F8	t8, huit-pièces	1	studio	0
marcher#VI.2	alimenter	1	fonctionner	1
se désintéresser	désintéresser	1	séparer	0
godasses	tatanes, grolles, pompes, chaussures	1	chaussure	0
mécanicienne#II	mécano	1	mécanicien	0
mastiquer	mâcher	1	broyer	0
sept-pièces	f7, t7	1	appartement	0
cradingue#VII	cradoque, crado, cracra, crasseux, craspec, crade	1	décalé	0
épine#2	aiguille	1	arbuste	0
crotte#I	étron, caca, merde	1	croquette	0
crade#I.1	cradoque, crado, cracra, crasseux, cradingue, craspec	1	con	0
cabochard_Adj	têtu	1	beau	0
s'accaparer#I	accaparer	1	voler	0
fesse#III	cul	1	femme	0
enceinte_Adj	en cloque, grosse	1	enceint	0
VTC#2	vélo tout chemin	1	vtt	0
impec_Adv	impeccablement	1	impeccable	1
rouge#VI.3b	communiste, coco	1	jaune	0
dégueu#V.2	dégueulasse	1	mauvais	1
thorax#a	cage thoracique	1	dos	0
collabo_N#a	collaborateur	1	résistant	0
bicross#1	bmx	1	cross	0
heuchère	désespoir des peintres	1	rose	0
dent_1#I.a	quenotte, ratiche	1	de	0
narine#2	évent	1	na	0
garé	stationné	1	gar	0
bouffer_1#V.2	se manger, manger	1	rendre	0
dégueulasse#I.1	dégueu	1	mauvais	1
gaviot	kiki, gorge, gargamelle	1	chapeau	0
bifteck#I	steak	1	hamburger	0
crade#VI.2	crado, crasseux	1	dégueulasse	1
tifs	chevelure, cheveux	1	doigt	0
gogues#b	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	couloir	0
T.G.V.	train à grande vitesse	1	japonais	0

alimenter#II	marcher	1	aliment	0
interviewer_N	intervieweur	1	interviewé	1
mécanicienne#I.1	machiniste	1	opérateur	0
guibolles	cannes, pattes, gambettes, jambes	1	main	0
grand-mère#II	vieillarde, mémé, mamie	1	mère	0
thorax#a	cage thoracique	1	corps	0
dégueu#II.2	dégueulasse	1	merde	0
craspec#V.3	cradoque, crado, cracra, crasseux, cradingue, crade	1	chaud	0
parterre#I	sol	1	par	0
W.-C.#a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	toilette	0
s'embarquer#I	embarquer	1	être	0
tapette#III	tata, folle, tante, tantouze	1	tape	0
zique	musique, zizique	1	z	0
crasseux#I	cradoque, crado, cracra, cradingue, craspec, crade	1	crasse	1
caille-lait	gaillet	1	thym	0
blatte	cafard	1	poule	0
crasseux#III.2	crado, crade	1	crasse	1
emprunteuse	prêteur, emprunteur, prêteuse	1	emprunt	1
cradingue#VI	cradoque, crado, cracra, craspec, crade	1	ridicule	0
congruent#I	adéquat, idoine, conforme, adapté, approprié	1	pertinent	0
s'appliquer#I	appliquer	1	convenir	0
bénard	falzar, fendard, froc, ben, pantalon, futal, fute, culotte	1	chapeau	0
s'éveiller	se réveiller	1	réveiller	0
patronyme#1	nom de famille, nom	1	portait	0
cradoque#I.1	crado, cracra, crasseux, cradingue, craspec, crade	1	rouge	0
FCPR	fonds commun de placement à risque	1	fc	0
décélérer#I	ralentir	1	déc	0
nourrice	mère nourricière	1	mère	0
se ramasser#II.2	ramasser, attraper, choper, prendre, se manger, se prendre, recevoir	1	ramer	0
enceinte_Adj	en cloque, grosse	1	enceint	0
crade#VI.1	cradoque, crado, cracra, crasseux, cradingue, craspec, crade	1	dégueulasse	0
cradoque#VI.3	crado, cracra, crasseux, cradingue, craspec, crade	1	de	0
enceinte_Adj	en cloque, grosse	1	prisonnier	0
crasseux#II	cradoque, crado, cracra, cradingue, craspec, crade	1	honteux	0
crade#V.2	craspec	1	cra	0
bifteck#I	steak	1	hamburger	0
cracra#II	cradoque, crado, crasseux, cradingue, craspec, crade	1	cra	0
menotte_2	mimine	1	tête	0
COB	commission des opérations de bourse	1	c	0
chiottes#I.a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, tinettes, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	toilette	0
enceinte_Adj	en cloque, grosse	1	ensemble	0
ciboulot	tête, cerveau, cafetière, matière grise, cervelle, caboche, cigare, crâne	1	ventre	0

TABLEAU A.9 – Évaluation des candidats absents du DS 0 avec un score normalisé = 1 (méthode <SEP>)

Entrée	**Lexs.**	**Score**	**Candidat**	**Qsyn**
occurrence	cas	1	occasion	1
absorbeur	chaussette	1	écran	1
s'étioler	dépérir	1	s	0
toasté	grillé	1	to	0
entourer#5	choyer	1	être	0
pères#II.2	ancêtres, aïeux	1	parent	1
parer	éviter, chasser les mouches, se protéger	1	porter	0
frasque	fantaisie, excentricité	1	aventure	0
piquet	sardine, pic	1	poteau	1
soigneux	minutieux, méticuleux	1	soigné	1
s'écouler#II	passer	1	être	0
horizontalement	couché	1	horizontal	1
mobiler_N#II	ameublement, meuble	1	mobilier	0
partir_2	partager	1	séparer	1
typhon	cyclone, tornade, vent	1	inondation	0
s'enfouir	s'enfoncer	1	dormir	0
annihiler	anéantir	1	détruire	1
s'énerver	sortir de ses gonds, voir rouge	1	énerver	0

shampouineuse	coupe-tif, coiffeuse	1	prostitué	0
s'allonger_1#II.1	grandir	1	croître	1
bouffer_1#III.2	accaparer, ronger, consumer	1	manger	0
appuyer#I.3	placer	1	poser	1
crado#III	cradoque, sale, cracra, dégueu, dégueulasse, crasseux, cradingue, crade	1	cra	0
s'écarter#II	s'éloigner	1	écarter	0
rabattable	pliable, pliant	1	rabat	0
articuler_2#a	articuler, prononcer	1	écrire	0
commémorer#2	commémorer, célébrer	1	rappeler	1
s'arrêter#I.1	s'immobiliser, piquer du nez	1	arrêter	0
lilas#II	violet	1	blanc	0
craquelure	cheveu	1	trou	0
imam	religieux	1	<unk>	0
T3	trois-pièces, f3	1	t	0
aspirant	aspirateur	1	</s>notused	0
quart-temps	période	1	quart	1
F5	t5, cinq-pièces	1	studio	1
fausseté	inexactitude	1	faux	1
transformable	modifiable	1	transformé	1
futal	falzar, fendard, froc, ben, bénard, pantalon, jean, fute, corsaire, culotte	1	blouson	0
s'enfoncer#I.2	s'enfouir, se tasser	1	enfoncer	0
dormir#I.1a	en écraser, sommeiller, s'allonger, roupiller, pieuter, somnoler, siester, dormir, ronfler, se reposer, pioncer	1	être	0
lune#I.1a	lune	1	l	0
valet#I	domestique, valet de chambre, servant	1	compagnon	0
BMX#1	biclou, biclo, bécane, bicross, vélocipède, bicyclette, vélo, petite reine	1	ligne	0
sous-ventrière	sangle	1	bride	0
heurter#III	vexer	1	déranger	1
museau#I.1b	gueule	1	yeux	0
cheveu#I.1	poil	1	cheveux	0
vermine#I	insectes	1	ver	0
trottiner#1	trotter	1	courir	1
pagaille#I	bordel, souk, désordre	1	masse	0
s'arrêter#I.1	s'immobiliser, piquer du nez	1	être	0
se laver#II	se débarrasser	1	sortir	0
binouze	canette, pression, bière, demi, mousse	1	autre	0
réserver#II	préparer, destiner	1	faire	0
water-closet#a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, latrines, chaise percée, tinettes, urinoir, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	sanitaire	1
cation	ion	1	gaz	0
félidés	chat, félinés	1	animal	1
braire#I	crier	1	broder	0
départ#V	naissance	1	point	0
flacon#a	gel douche, fiole	1	bouteille	1
s'allonger_2#I.b	se coucher	1	être	0
se pinter	boire, picoler	1	p	0
blaireau_2	imbécile	1	mec	0
lieu-dit	hameau	1	croisement	0
promeneur	flâneur	1	randonneur	1
hélicoïde	spirale	1	axe	0
minou#II	moule, tête à claques, con, sexe, foufoune, vulve, zézette, chatte, abricot	1	clitoris	1
friterie	baraque à frites	1	restaurant	1
hydrolat	eau	1	savon	0
vivre#II.2	nager, traverser	1	voir	0
grands-parents	bon-papa, papi, bonne-maman, mémère, mère-grand, mémé, grand-mère, grand-papa, grand-maman, pépé, mamie, aïeule, ancêtre, grand-père, aïeul, pépère	1	enfant	0
cradoque#VI.2	glauque	1	sombre	0
amical#I.2	cordial, chaleureux, ami	1	regard	0
ratiboiser	raser, tondre	1	mordre	0
allonger_1#II.1	agrandir, rallonger	1	compléter	0
rayonnage	étagère	1	rayon	0
petite-nièce	arrière-petite-nièce	1	fille	0
lessive#I	lavage	1	vaisselle	0
ressasser	repéter, redire	1	répéter	0
émerger#II	sortir	1	avoir	0
huppé#II	riche	1	populaire	0
W.-C.#a	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, latrines, chaise percée, tinettes, urinoir, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	wc	0
gravitation	attraction	1	gravit	0
hâtif#III	précoce	1	élevé	0

allonger_1#II.2a	rallonger	1	augmenter	0
demi-tour	volte-face	1	demi	0
jazzy	jazzique, jazzistique	1	jazz	1
casquette_1#II	rôle, costume	1	chapeau	1
laver#V.2	peindre	1	être	0
régate#II	cravate	1	couronne	0
touffe	flocon, brosse	1	frange	0
s'endormir#III	décéder, tuer, périr, emporter, mourir, trépasser, disparaître, clamser, succomber, perdre la vie, casser sa pipe, s'éteindre, crever, passer l'arme à gauche, expirer	1	dormir	0
s'emporter	se fâcher, monter sur ses grands chevaux	1	emporter	0
bougre	animal	1	gars	1
transpercer#II	traverser	1	tomber	0
s'ouvrir#I	s'écarter	1	ouvrir	0
dégueu#V.2	dégueulasse	1	mauvais	1
malpropre#V.2	cracra, sale	1	mal	0
asseoir#III	baser, fonder, calculer	1	appuyer	1
strident	aigu	1	profond	0
cracra#II	cradoque, sale, crado, dégueu, dégueulasse, crasseux, cradingue, craspec, crade	1	cra	0
tifs	tête, poil, chevelure, crinière, tignasse, cheveux, cheveu	1	pull	0
circuit#1	boucle, tour	1	voiture	0
incarnat#b	rouge	1	orange	1
s'agiter#II	bouger	1	s	0
ronfler#III	ronronner	1	tourner	0
avachi	tassé, affalé, vautré	1	assis	0
babine#1	lèvres, babines	1	bouche	0
turlute	fellation, pompier, gorge profonde, gâterie, pipe	1	plaisanterie	0
parterre#II.3	champ	1	tapis	0
clic-clac	convertible, canapé-lit	1	clic	0
avocate#I	plaideuse, défenseuse, avocassier	1	avocat	0
chipoter#I	bouffer, se bouffer, manger, picorer, mangeotter, se manger	1	prendre	0
considérer#I	observer	1	regarder	1
sympathiser	faire ami, briser la glace	1	collaborer	0
avaler#IV	bouffer, manger	1	aval	0
fuyard_Adj	fugitif	1	disparu	0
surnager#I.2	nager, flotter	1	surgir	0
encadrer#VI.1	diriger	1	cadrer	0
ciseau#1	burin	1	mortier	0
débrouillardise	astuce	1	débrouille	1
congruent#II	égal	1	parallèle	0
se sacrifier	se dévouer	1	être	0
avaler#I.1	ingérer, bouffer, se bouffer, manger, boire, avaler, aspirer, se manger	1	aval	0
geyser#II.1	flot, torrent, fleuve, coulée, rivière	1	nuage	0
survêt	jogging, survêtement	1	vêtement	1
similitude	analogie, point commun, ressemblance, similarité, identité	1	différence	0
raccourcir#I	réduire, diminuer	1	prolonger	0
criterium	course, tournoi	1	championnat	1
crottin	fumier	1	...	0
remuable	déplaçable	1	modifiable	0
entassement#I.b	amoncellement, amas, tas, colline	1	assemblage	0
bouffer_1#III.1	ronger, éroder, se bouffer, manger, creuser, grignoter, détruire, attaquer	1	mordre	0
évaluable	calculable	1	évalué	1
crado#IV	cradoque, sale, crado, cracra, dégueu, dégueulasse, salaud, cradingue, craspec, crade	1	cra	0
atome#I.2	atomes crochus, élément, particule	1	gaz	0
s'ébahir	stupéfier, stupéfaire, ébahir, surprendre, étonner	1	parler	0
arpion	ripatons, panards, pieds, pince	1	poumon	0
analogue_N#b	équivalent	1	dérivé	0
s'ouvrir#I	s'écarter	1	ouvrir	0
ermitage	désert	1	église	0
navet#II	film	1	drame	0
cactus	figuier de barbarie	1	plante	1
bassesse	vilenie	1	bêtise	0
évanouissement#II	disparition	1	retrait	0
meunerie	moulin	1	me	0
crado#VI.1	cradoque, crado, cracra, cradingue, craspec, crade	1	cra	0
kiwi_1#II	actinidia	1	arbre	0
évènement#I	évènement historique, fortune, coup d'éclat, phénomène	1	événement	0
tambouriner	frapper	1	tambour	0
plumer	escroquer, voler, flouer, arnaquer, rouler dans la farine, rouler, pigeonner, blouser, gruger	1	ruiner	0
vanter	louer	1	chanter	1
mordiller#I	bouffer, se bouffer, manger	1	mordre	0
choisir#2	prendre	1	être	0

grignotage	chipotage	1	découpage	0
complétude	exhaustivité	1	qualité	0
lessive#III	linge	1	vaisselle	0
remédier#I.a	guérir	1	réagir	0
excentricité_2#I.1a	distance, écartement, écart	1	...	0
lien#II.2	rapport, relation	1	amoureux	0
rouler_1#VI.3	se déplacer	1	roul	0
contorsion#I	gymnastique	1	mouvement	1
brosser_1#I	frotter	1	dresser	0
symétrique#II	réciproque	1	absolu	0
hold-up#II	attaque, vol, appropriation	1	meurtre	0
aigreur#I	amertume	1	douceur	0
là-dedans#I	là	1	dedans	0
génital	reproducteur	1	sexuel	1
cloître	désert	1	monastère	1
commérage	clabaudage, ragot, racontar, potin, bavardage	1	commentaire	0
confectionner	élaborer, tailler	1	confection	1
vermine#II	insecte	1	peste	1
linéaire	droit	1	carré	0
compatir	s'apitoyer, plaindre	1	compa	0
se farder	se maquiller	1	maquiller	0
analogie#c	similitude, ressemblance	1	</s>notused	0
volley-ball#2	partie, match, volley, volley de plage, beach-volley	1	foot	0
geyser#II.1	flot, torrent, fleuve, coulée, rivière	1	tourbillon	0
majorer	accroître	1	augmenter	1
eldorado	fleuve de lait et de miel, paradis, pays de cocagne	1	enfer	0
installer#I.1	loger	1	mettre	0
gravitation	attraction	1	gravit	0
riot	ruisseau	1	chariot	0
délabré	endommagé, détérioré	1	dé	0
s'atteler	entreprendre	1	participer	0
cradoque#V	dégueulasse, salaud, dégueu, sale	1	horrible	1
arrière-neveu	neveu	1	arrière	0
kidnapper#II.2	monopoliser, accaparer, s'approprier, s'accaparer	1	voler	0
pelage_1	manteau, fourrure, robe	1	poil	1
vélodrome	piste	1	stade	0
voler_1#IV.a	voleter, se déplacer	1	tomber	0
réprimande	savon, volée de bois vert, carton rouge, leçon	1	réplique	0
étalage#I.2a	collection	1	éventail	1
défaillir	s'évanouir, tomber dans les pommes, tourner de l'œil	1	de	0
sainement	sain	1	saine	0
raclée#II	défaite	1	claque	1
porte-documents	serviette, attaché-case, mallette	1	sac	1
courir#VI	s'exposer	1	représenter	0
marquant	frappant	1	emblématique	1
cogner#II.2	heurter	1	tourner	0
tante#I	tata, tatie, tantine	1	mère	0
louvoyer#I.2	naviguer, se déplacer, zigzaguer, circuler	1	passer	1
chaussette#II.2	toboggan, tube	1	chaussure	0
train-train	routine, vie, chemins battus	1	train	0
voyou_N#1	criminel, malfaiteur, délinquant, racaille, arsouille, malfrat, loubard, bandit, gangster	1	voleur	0
s'amuser#III	occuper, s'occuper	1	jouer	1
battre_2	parcourir	1	à	0
inquiéter#II	tracasser	1	inquiet	1
contrôlable#II	maîtrisable	1	fiable	0
W.-C.#b	chiotte, waters, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, latrines, chaise percée, tinettes, urinoir, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	c	0
wawas#a	waters, chiotte, pipi-room, tartisses, commodités, goguenots, cabinet d'aisances, w.-c., chiottes, gogues, sanitaires, cabinets, latrines, chaise percée, tinettes, urinoir, tinette, petit coin, wawas, toilettes, lieux d'aisances, water-closet, cagoinces, petit endroit, vécés	1	w	0
s'endimancher	s'habiller	1	courir	0
succomber#II	céder	1	résister	0
courber#2	rouler, se pencher	1	plier	1
crado#V.2	sale, dégueu, dégueulasse, crasseux, crade	1	cra	0
kidnapper#I	enlever	1	capturer	1
contourner	éviter	1	traverser	0
bourré#I	gonflé, surchargé	1	rempli	1
arrière-train#2	siège, fesses, cul, séant, paire de fesses, postérieur, derrière, popotin, lune, derche, miches, pétard, fessier, croupion, croupe, panier	1	arrière	0
lessive#II.2	solution	1	bicarbonate	0

s'amonceler	s'entasser, s'accumuler, s'amasser	1	s	0
tortue#I	tortue de mer, tortue marine	1	créature	0
s'intégrer#II.2	intégrer	1	s	0
blanchisserie#1	teinturerie, atelier	1	fabrique	0
voler_1#V	pleuvoir, voleter	1	vol	0
boiter#II.2	clocher, fonctionner	1	boire	0
ivre#I.2	rempli	1	épris	0
cheveu#I.1	poil	1	cheveux	0
dépotoir#I	décharge	1	atelier	0
fumier#II	vache, ordure, salaud, connard, pourriture	1	cochon	1
moineau#II	zigounette, zizi, biroute, verge, teub, bite, pénis, membre, phallus, queue, membre viril, quéquette, bistouquette, pine, braquemart, zob, manche, sexe, robinet, nouille, chibre, cigare	1	moi	0
allonger_1#IV	tirer	1	faire	0
allonger_2#IV	régler, donner, payer, abouler	1	prolonger	0
fesses#I.a	siège, cul, paire de fesses, séant, postérieur, derrière, popotin, derche, lune, miches, pétard, fessier, croupion, arrière-train, croupe, ballons, panier	1	cuisse	0
lave-phare	essuie-glace, appareil	1	phare	0
répandre#I.2	provoquer	1	présenter	0
croupe#I	arrière-train, fesses	1	cuisse	0
pasteure	femme pasteur	1	pasteur	0
s'engager#III.2b	entrer	1	s	0
adorateur#II.1	fan	1	ado	0
rhino	rhinocéros	1	rhin	0
s'allonger_2#I.a	se coucher	1	aller	0
cogner#I.1	battre, frapper	1	c	0
crotte#I	matière fécale, fèces, selle, mince !, étron, merde, caca, cigare, excréments, colombin, bol fécal	1	croquette	0
mettre#I.2a	passer, revêtir, endosser, se coiffer, enfoncer, enfiler, remettre	1	porter	0
quai#II.1	wharf	1	qu	0
pépé#I	bon-papa, bonne-maman, mémère, mère-grand, grand-mère, mémé, grand-papa, grand-maman, pépère, grands-parents, mamie, aïeule, grand-père, ancêtre, aïeul, papi	1	pé	0
employer#II	embaucher, recruter	1	...	0
plus-value	bénéfice	1	opération	0
nager#II.2	baigner, se trouver, croupir, accueillir, tremper	1	passer	0

TABLEAU A.10 – Évaluation des candidats absents du DS 1 avec un score normalisé = 1 (méthode <SEP>)

Annexe B

Scripts

Listing B.1 – Module de lexicalisation souple à partir du RL-fr

```
"""
Imports a lexical network and creates a semanticon out of it.
"""
import lexnet
import argparse
from sortedcontainers import SortedDict
import pandas as pd
import re
import warnings


DEFAULT_MAX_APPROX = 0
DEFAULT_PATH = '../lng/fre/LN'    # default path for input files
DEFAULT_FILE = './semanticon.dict'  # default output file
DEFAULT_SEPARATOR = '\t'          # default separator for csv files
DEFAULT_ENCODING = 'utf-8'        # default character encoding (for input and output)

def expand_synonyms(semanticon, max_approx):
    for entry, lexs in semanticon.items():
        syns = get_syns(semanticon, entry, max_approx, visited=[], n_approx=0)
        lexs.update(syns)
    return semanticon

def get_syns(semanticon, entry, max_approx, visited, n_approx):
    syns = set()
    lexs = semanticon[entry]
    visited.append(entry)
    for lex, approx in lexs:
        sum_approx = approx + n_approx
        if lex not in visited and sum_approx <= max_approx:
            result = (lex, sum_approx) # connaître la profondeur de synonymie
            syns.add(result)
            syns.update(get_syns(semanticon, lex, max_approx, visited, sum_approx))
```

```
    return syns

def build_semanticon(ln, max_approx, same_combin, file=DEFAULT_FILE,
    encoding=DEFAULT_ENCODING):
    """Builds and writes a semanticon from a Lexical Network"""

    print(f'\x1b[0;34mBuilding␣semanticon\x1b[0m')

    # Reindex nodes
    nodes = ln['nodes']

    # Trivial lexicalization mappings
    # We use SortedDict so that the items will stay sorted by key (for more readable output)
    # Lexicalization is a tuple (lexical unit, approx, Triv) where approx flags approximate
        lexicalizations and triv indicates the trivial lexicalization type
    semanticon = SortedDict({std_name:set(((std_name, 0, 'Triv'))) for std_name in
        nodes.std_name})

    # Add synonyms and derivational LFs
    lfs = ln['lfs']
    deriv_lfs = ln['deriv_lfs']
    lf_names = ln['lf_names']
    if same_combin:
        lfs['same_combin'] = lfs.lf_id.apply(lambda x: deriv_lfs.loc[x, 'same_combin'] if x
            in deriv_lfs.index else None)
        filter = lfs.lf_id.isin(deriv_lfs.index) & lfs.same_combin
    else:
        filter = lfs.lf_id.isin(deriv_lfs.index)
    syns = lfs[ filter ]
    # keep lex-lf_name mapping
    syns['lf_name'] = syns.lf_id.apply(lambda x: lf_names.loc[x, 'lf_name'])
    # Approx synonym lfs
    syns['approx'] = syns.lf_id.apply(lambda x: deriv_lfs.loc[x, 'approx'])
    #synonym dictionary
    for lf_id, row in syns.iterrows():
        if (deriv_lfs.loc[row.lf_id].type == "paradigmatic" or row.merged) and not '~' in
            str(row.frame):
            # source -> target
            if row.approx <= max_approx:
                lex = (nodes.loc[row.target_id].std_name, row.approx, row.lf_name)
                entry = nodes.loc[row.source_id].std_name
                semanticon[entry].add(lex)
                # target -> source (Syn is reciprocal)
                lex = (nodes.loc[row.source_id].std_name, row.approx, row.lf_name)
                entry = nodes.loc[row.target_id].std_name
                semanticon[entry].add(lex)

    expand_synonyms(semanticon, max_approx=max_approx) # returns new semanticon with
        updated syns

    d = []
    for k, i in semanticon.items():
```

```
        for t in i:
            d.append({'std_name' : k, 'syn' : t[0], 'depth' : t[1]})
    syns_df = pd.DataFrame(d)
    nodes_names = nodes.reset_index().set_index('std_name')
    syns_df['node_id'] = syns_df.std_name.apply(lambda x: nodes_names.loc[x, 'node_id'])
    syns_df.set_index('node_id', inplace=True)

    # Write to file
    with open(outfile, 'w', encoding=encoding) as f:
        f.write(f'// This file was generated automatically\n\n')
        f.write('semanticon {\n')
        n = 0 # counter for nodes
        l = 0 # counter for lexicalizations
        for sem, lexs in semanticon.items():
            just_lexs = set()
            just_qlexs = set()
            for lex, a, lf in lexs:
                if a == 0:
                    just_lexs.add(lex)
                else:
                    just_qlexs.add(lex)
            written_lexs = []
            written_qlexs = []
            for lex in just_lexs:
                written_lexs.append(f'lex={lex}')
            for lex in just_qlexs:
                written_qlexs.append(f'qlex={lex}')
            all_lexs = written_lexs + written_qlexs
            f.write(sem)
            f.write(' {')
            f.write(' '.join(lex for lex in sorted(all_lexs)))
            f.write('}\n')
            n += 1
            l += len(all_lexs)
    print(f'Wrote {l} lexicalizations for {n} semantemes in {outfile}')
    if n != len(ln['nodes']):
        print(f"\x1b[0;31mWarning:\x1b[0m The number of semantemes doesn't match the number
            of nodes")

    return syns_df, semanticon
```