Construction automatique de lexiques thématiques

Suzanne Mpouli*, Christine Largeron*, Michel Beigbeder**

*UJM-Saint-Etienne, Laboratoire Hubert Curien UMR 5516, F-42023, France
Suzanne.Mpouli@univ-st-etienne.fr, Christine.Largeron@univ-st-etienne.fr
**Univ Lyon, IMT Mines Saint-Etienne, Institut Henri Fayol,
Univ Jean Monnet, IOGS, CNRS, LHC, F - 42023 Saint-Etienne FRANCE
michel.beigbeder@emse.fr

Résumé. Nous présentons *Lexifield*, un système entièrement automatique et indépendant de la langue pour la construction de lexiques spécifiques à un domaine à partir de courtes listes de termes [1]. *Lexifield* s'appuie sur un modèle de plongement lexical (*word embedding*), un dictionnaire de définitions et un dictionnaire de synonymes. Pour évaluer *Lexifield*, quatre lexiques ont été générés : un lexique en français pour le sujet "son" et trois lexiques en anglais pour les sujets "sound", "taste" et "odour". Les résultats ont confirmé ses bonnes performances par rapport à d'autres systèmes de l'état de l'art.

1 Introduction

La création manuelle de lexiques est non seulement longue et coûteuse mais peut également aboutir à un contenu qui n'est pas à jour. Notre objectif est donc de construire automatiquement des lexiques, au sens original du terme, puisque nous voulons générer des listes de mots qui couvrent un sujet spécifique : trois des cinq sens humains (odorat, ouïe et goût) dans notre étude de cas, bien que notre méthode puisse également être appliquée à d'autres sujets.

Le même problème a été abordé par Tekiroglu et al. (2014) qui génère Sensicon, un lexique se rapportant aux cinq sens à partir d'une liste de 277 lexèmes sélectionnés manuellement dans FrameNet (Baker et al. (1998)) et étendue automatiquement avec WordNet (Fellbaum (1998)). Enfin, les voisins des mots appartenant à la liste filtrée sont récupérés dans un grand corpus. Ces mots candidats sont classés à l'aide de la mesure NPMI (*Normalized Pointwise Mutual Information*) pour obtenir une liste finale triée.

Une approche similaire est décrite dans Riloff et Shepherd (1997) qui utilise uniquement une courte liste de mots relatifs au sujet sélectionné et un corpus lié au domaine pour étendre celle-ci. Les mots qui co-apparaissent dans une phrase avec les mots initiaux sont ordonnés en fonction de leur fréquence d'occurrence avec les mots d'origine et de leur fréquence d'apparition dans l'ensemble du corpus. Ce système a été amélioré par Roark et Charniak (1998) où la connaissance syntaxique est utilisée pour éliminer du bruit dans la liste des résultats. Le système Empath (Fast et al. (2016)) est similaire, mais utilise des technologies plus récentes telles qu'un modèle de plongement de mots entrainé sur le corpus Wattpad [2] et des outils de

1. Ce travail a été réalisé dans le cadre du projet SoundCITYve soutenu par le Labex IMU
2. `http://wattpad.com`

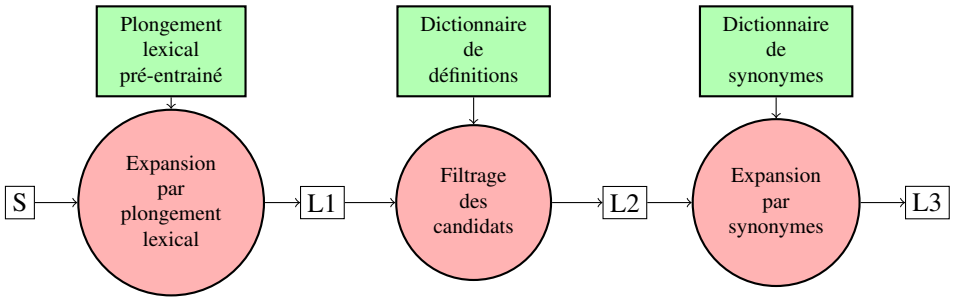

FIG. 1 – *La méthode proposée. S : liste initiale (les graines). L1 : liste des mots voisins dans le plongement des mots de S. L2 : liste des mots après le filtrage par le dictionnaire. L3 : liste des mots après expansion avec le dictionnaire de synonymes.*

traitement automatique de la langue pour filtrer les mots vides et lemmatiser les mots simples avant l'apprentissage. Enfin, Conceptvector (Park et al. (2018)) un autre système basé aussi sur le plongement de mots, se veut plutôt un outil interactif d'assistance à la construction de lexiques.

De ces travaux, nous avons retenu l'idée d'utiliser une liste initiale de mots que nous appelons des graines pour lancer le processus ; cependant, dans notre système la taille de cette liste est très réduite (entre trois et six mots), comparativement par exemple aux 277 lexèmes considérés dans Tekiroglu et al. (2014). Comme dans le système Empath (Fast et al. (2016)), nous considérons que les modèles de plongement de mots peuvent être très utiles pour étendre cette première liste de termes, mais qu'ils risquent aussi de produire des termes inappropriés qui doivent être enlevés. Aussi, dans notre système, ces mots sont-ils ensuite filtrés à l'aide de ressources linguistiques, rendant Lexifield complètement automatique alors que les systèmes mentionnés précédemment requièrent l'intervention de l'utilisateur. Notre système s'appuie donc sur un modèle de plongement de mots (*word embedding*), un dictionnaire de définitions et un dictionnaire de synonymes. Il a été testé sur deux langues : anglais et français, et les expériences confirment son intérêt. Dans la section 2, nous expliquons en détail toutes les étapes de notre processus d'extension de la liste de mots clés. La section 3 présente les différentes expériences que nous avons effectuées en anglais et en français, et les résultats sont détaillés dans la section 4.

2 Le système Lexifield

Le système *Lexifield*, que nous avons développé, construit automatiquement un lexique spécifique à un domaine, à partir de quelques mots-clés entrés par l'utilisateur pour décrire le sujet qui l'intéresse. La méthode sous-jacente repose sur trois hypothèses :

— Un lexique spécifique à un domaine doit contenir des mots synonymes ou sémantiquement proches.
— Comme les mots du lexique sont sémantiquement liés, ils doivent apparaitre dans les définitions du dictionnaire d'autres mots appartenant à ce lexique.

```
Données :
  S : ensemble des mots graines
  E : plongement lexical
  Def : dictionnaire de définitions
  Syn : dictionnaire de synonymes
  α : rayon
  β : seuil de Jaccard
Résultat :
  L : lexique étendu
// Expansion basée sur le plongement lexical (étape 1)
C_E = ∅;
pour s ∈ S faire
    E_s = {w; cos(E(s), E(w)) > α};
    C_E = C_E ∪ E(s);
fin
L1 = C_E ∪ S;
// filtrage des candidats (étape 2)
L = S;
répéter
    stop = TRUE;
    pour c ∈ C_E faire
        si Def(c) ∩ L ≠ ∅ alors
            L = L ∪ {c} ;
            C_E = C_E\{c} ;
            stop = FALSE;
        fin
    fin
jusqu'à stop = TRUE;
L2 = L;
// Expansion basée sur les synonymes (étape 3)
LC = ∅;
répéter
    pour m ∈ L faire
        LP_m = ∅;
        // LP_m est utilisé pour construire LP(m, L)
        pour chaque sens i de m faire
            si Syn(m, i) ∩ L\{m} ≠ ∅ alors
                LP_m = LP_m ∪ Syn(m, i);
            fin
        fin
        pour chaque w de LP_m\L faire
            LP_w = ∅ ;
            // LP_w est utilisé pour construire LP(w, LP(m, L))
            pour chaque sens j de w faire
                si
                Syn(w, j) ∩ LP_m\{w} ≠ ∅
                alors
                    LP_w =
                    LP_w ∪ Syn(w, j);
                fin
            fin
            si |LP_m ∩ LP_w|/|LP_w| ≥ β
            alors
                LC = LC ∪ {w};
            fin
        fin
    fin
    L3a = L2 ∪ LC;
    L' = {w ∈ LC; Def(w) ∩ L ≠ ∅};
    L = L ∪ L';
    L3b = L ∪ L';
jusqu'à end_condition;
```

FIG. 2 – *Algorithme*

— Les mots qui sont synonymes ou proches sémantiquement sont aussi plus susceptibles d'être proches dans l'espace de plongement de mots.

Compte tenu de ces hypothèses, la méthode de construction de lexiques que nous proposons comporte trois étapes, comme le montre la Figure 1. Tout d'abord, le système sélectionne, à partir des mots clés fournis par l'utilisateur, des termes candidats dans l'espace de plongement de mots. Ensuite, il supprime les termes non pertinents en utilisant un dictionnaire de définitions. Enfin, il ajoute les synonymes les plus pertinents des termes restants à l'aide d'un dictionnaire de synonymes. Le système fait donc appel à trois ressources lexicales et peut ainsi créer un lexique spécifique à un domaine pour n'importe quelle langue pour laquelle de telles ressources sont disponibles. Les différentes étapes de ce processus sont détaillées ci-dessous et dans l'algorithme.

2.1 Expansion par plongement de mots (cf. Algorithme - Étape 1 , Fig. 2)

Comme indiqué précédemment, dans la première phase du processus, l'utilisateur propose des mots-clés liés à un domaine. Dans le reste de l'article, cet ensemble de graines sera noté S. En pratique, entre deux et cinq mots sont suffisants pour cibler le sujet du lexique. L'utilisateur doit aussi fournir un modèle de plongement de mots tel que ceux introduits par Mikolov et al. (2013), Levy et Goldberg (2014) ou Globerson et al. (2007). Ces modèles permettent de coder les mots d'un corpus sous forme de vecteurs dans un espace vectoriel. Dans la suite, E désigne ce modèle de plongement de mots et $E(w)$ représente le vecteur associé à un mot particulier w dans l'espace vectoriel $\mathbb{R}^p$. Il est clair que selon notre troisième hypothèse, les modèles de plongement de mots retenus dans notre système sont ceux où les mots apparaissant dans un même contexte et donc ayant le plus de chances d'avoir le même sens seront associés à des vecteurs proches dans l'espace vectoriel (Lavelli et al. (2004)).

Afin d'élargir la liste S des mots de départ, la première étape consiste à interroger le modèle de plongement de mots E pour acquérir les mots proches de chaque mot-clé dans l'espace de représentation et ensuite à chercher des synonymes de ces nouveaux mots. Il convient de noter que nous ne procédons pas comme dans Empath (Fast et al. (2016)) en ajoutant les vecteurs de tous les mots de départ pour trouver les voisins de leur somme. Au lieu de cela, nous considérons chaque graine séparément et retenons ses voisins dans l'espace vectoriel. Plus formellement, à partir de S, l'ensemble des termes de départ, pour chaque terme $s \in S$, différentes stratégies peuvent être appliquées pour définir l'ensemble E_s de ses voisins. La première peut être de sélectionner les k vecteurs les plus proches de $E(s)$ dans E alors que la seconde consiste à choisir les mots associés aux vecteurs pour lesquels la similarité cosinus est supérieure à un seuil prédéfini α. Enfin une troisième option combine les précédentes et vise à retenir les k mots les plus proches parmi les vecteurs pour lesquels la similarité cosinus est supérieure à α. Ainsi, l'ensemble des termes candidats C_E est défini par : $C_E = \cup_{s \in S} E_s$.

2.2 Filtrage des candidats (cf. Algorithme - Étape 2, Fig. 2)

Le but du deuxième module, présenté à l'étape 2 de notre algorithme, est d'éliminer les mots qui ne sont pas pertinents dans l'ensemble des candidats afin d'obtenir un premier lexique L. Cette étape de filtrage est basée sur un dictionnaire sélectionné par l'utilisateur et exploite la deuxième hypothèse selon laquelle les mots appartenant au lexique doivent figurer dans les définitions du dictionnaire des autres mots de ce lexique car ils sont sémantiquement liés. Plus formellement, l'extraction de la liste de mots filtrés L est réalisée de la façon suivante :

Étant donné un candidat $c \in C_E$ et $\mathcal{D}ef(c)$ sa définition dans le dictionnaire, c est ajouté à L si $\mathcal{D}ef(c) \cap L \neq \varnothing$. Cette étape est répétée jusqu'à ce qu'il n'y ait plus de candidat.

2.3 Expansion basée sur des synonymes (cf. Algorithme - Étape 3, Fig. 2)

Ayant obtenu un premier lexique L de taille réduite, le dernier module vise à l'étendre en trouvant les synonymes des mots qu'il contient grâce à un dictionnaire de synonymes. Cependant, comme dans le module précédent, cette extension basée sur les synonymes nécessite un filtrage, en particulier à cause des mots homonymes (ayant la même orthographe mais des significations différentes) et polysémiques (ayant plusieurs significations). Par conséquent, pour désambiguïser les mots qui nous intéressent, un dictionnaire de synonymes qui différencie les

significations de chaque entrée de mot est requis. Ensuite, pour choisir le sens correct, nous émettons l'hypothèse que la liste de synonymes doit coïncider au moins partiellement avec notre liste de mots. Plus formellement, étant donné un mot $m \in L$ et $\mathcal{S}yn(m,i)$ la liste de ses synonymes pour le sens i, la liste $LP(m,L)$ des synonymes potentiels de m est générée de la façon suivante :

$$LP(m,L) = \{w; \exists i \, s.t. \;\; (w \in \mathcal{S}yn(m,i)) \wedge (\mathcal{S}yn(m,i) \cap L \backslash \{m\} \neq \varnothing)\}$$

De façon similaire pour un mot w appartenant à $LP(m)$ qui ne figure pas dans L, la liste de ses synonymes potentiels $LP(w, LP(m,L))$ est définie par :

$$LP(w, LP(m,L)) = \{w'; \exists j \, s.t. \;\; (w' \in \mathcal{S}yn(w,j)) \wedge (\mathcal{S}yn(w,j) \cap LP(m,L) \backslash \{w\} \neq \varnothing)\}$$

où j désigne un sens de w.

Ensuite, la liste des synonymes candidats $LC(m)$ de m est obtenue en filtrant la liste des synonymes potentiels LP à l'aide d'une mesure de Jaccard. Elle est définie par : $LC(m) = \{w \in LP(m,L); \frac{|LP(m,L) \cap LP(w, LP(m,L))|}{|LP(w, LP(m,L))|} \geq \beta\}$ où β est un paramètre dont la valeur détermine le niveau de pertinence requis. Ainsi, w est conservé lorsque son pourcentage de synonymes potentiels appartenant à $LP(m,L)$ est élevé.

La liste complète des candidats LC est égale à l'union des listes de candidats des mots appartenant à L : $LC = \cup_{m \in L} LC(m)$

Enfin, ces candidats sont filtrés comme expliqué précédemment en considérant leurs définitions. Soit un candidat $w \in LC$ et $\mathcal{D}ef(w)$ sa définition dans le dictionnaire, w est ajouté à L si $\mathcal{D}ef(w) \cap L \neq \varnothing$.

Il convient de noter que cette phase d'expansion peut être répétée plusieurs fois mais nos expérimentations ont montré qu'en général, un seul cycle était suffisant.

3 Protocole expérimental

3.1 Description de la tâche

Pour la partie expérimentale, nous avons travaillé sur quatre sujets : un en français ("son") et trois en anglais ("sound", "taste", "odour"). De plus, nous considérons également un dernier sujet correspondant à l'union de ces derniers ("taste" ∪ "odour"). À partir d'une liste contenant trois à six mots clés en fonction du sujet, le but de ces expériences est de construire le lexique spécifique à chacun des quatre concepts choisis. Les mots clés ont été choisis manuellement parmi les mots qui semblent refléter le mieux chaque sujet. Il s'agit de $\{son_{noun}, crier_{verb}, bruit_{noun}\}$ pour "son", $\{sound_{noun}, sound_{verb}, noise_{noun}\}$ pour "sound", $\{taste_{noun}, taste_{verb}, flavour_{noun}, flavour_{verb}, flavor_{noun}, flavor_{verb}\}$ pour "taste" et enfin $\{odour_{noun}, odor_{noun}, scent_{verb}, scent_{noun}, smell_{verb}, smell_{noun}\}$ pour "odour". Nous avons décidé de prendre en compte les orthographes britannique et américaine puisque le modèle de plongement de mots considèrent les deux. La liste pour le dernier sujet correspond à l'union des listes obtenues pour chacun des sujets "taste" et "odour".

3.2 Ressources lexicales

Comme indiqué dans la section précédente, la méthode proposée nécessite trois types de ressources : un dictionnaire de définitions (Def), un dictionnaire de synonymes (Syn) et un modèle de plongement de mots pré-entraîné (E). Wiktionary [3] étant disponible gratuitement dans diverses langues et basé sur des dictionnaires bien établis, il a été sélectionné comme dictionnaire de définitions pour cette étude à la fois pour les expériences en anglais et en français. Pour les deux langues, nous avons utilisé la collecte du 1er avril 2018 [4]. Les définitions ont été récupérées et nettoyées à l'aide d'expressions rationnelles.

Après avoir testé plusieurs dictionnaires en ligne, nous avons sélectionné `synonym.com` [5] pour l'anglais et *Dictionnaire électronique des synonymes* [6] pour le français.

Enfin, nous avons choisi le modèle de plongement de mots construit sur le Corpus TenTen français et anglais (Jakubíček et al. (2013)) en utilisant fastText (Bojanowski et al. (2017)) et le modèle SkipGram avec 100 dimensions [7] car ils ont été appris sur des corpus très volumineux et ont été lemmatisés ainsi qu'annotés en partie du discours. Le cosinus a été fixé à 0,5 et nous avons considéré pour k les 1000 premiers mots voisins dans l'espace vectoriel.

3.3 Évaluation : listes de référence, mesures et modèles de référence

Listes de référence : Nous considérons également trois ressources pour construire des listes de vérité terrain (*LGT*) utilisées pour évaluer les lexiques générés.

Linguistic Word Inquiry Count Comme dans les études antérieures de Fast et al. (2016), nous avons utilisé le *Linguistic Word Inquiry Count* (LIWC) (Pennebaker et al. (2001)) comme vérité terrain. Il a été créé manuellement et conçu pour analyser des textes en fonction de catégories psychologiques spécifiées. Pour cette expérience, nous avons utilisé la version 2007 dans laquelle il existe une catégorie combinant les cinq sens ("Percept") et une catégorie pour la vue ("See"), l'audition ("Hear") et le toucher ("Feel"). Pour créer la liste d'évaluation finale pour les sujets "taste" et "odour", nous avons retiré de la catégorie "Percept" les mots présents dans les trois catégories précitées. La liste sensorielle extraite de la LIWC 2007 contient 71 termes dont 55 sont des racines.

Le Thésaurus de Roget En plus du LIWC, nous avons basé notre évaluation sur une autre ressource souvent utilisée pour mesurer la similarité des mots, le Thésaurus de Roget [8] qui contient une sous section "Special Sensation" au sein de laquelle figurent les différents sens. Ce thésaurus nous a permis de construire trois listes de référence (vérité terrain) contenant 326 termes (*Head concepts* 390-397), 159 (*Head concepts* 398-401) et 512 (*Head concepts* 402-412) respectivement pour les sujets "taste", "odor" et "sound".

Les Verbes Français et **Le Dictionnaire Électronique des Mots** Pour le lexique français, nous nous sommes appuyés sur deux bases de données lexicales : *Les Verbes Français* (Dubois et Dubois-Charlier (1997)) et son pendant *Le Dictionnaire électronique des mots* (Dubois et Dubois-Charlier (2010)). La première donne des informations sur l'utilisation syntaxique et sémantique de verbes français, la seconde intègre aussi les adjectifs, les noms et les adverbes.

3. `https://www.wiktionary.org/`
4. `https://dumps.wikimedia.org/backup-index.html`
5. `https://www.synonym.com/`
6. `http://crisco.unicaen.fr/des/`
7. `https://embeddings.sketchengine.co.uk/static/index.html`
8. `http://www.gutenberg.org/ebooks/22`

Nous avons commencé par extraire les verbes relatifs au son. Ensuite, nous avons récupéré automatiquement dans *Le Dictionnaire électronique des mots* (Dubois et Dubois-Charlier (2010)) tous les adjectifs auxquels l'étiquette sémantique "adj bruit" a été attribuée ainsi que tous les noms qui soit appartiennent au champ lexical "voix, bruit", soit ou relèvent de l'une des sous-catégories des verbes précédemment extraits et sont liés sémantiquement aux termes "bruit", "dit", "crier", "cri", "dire" ou "son".

Le lexique considéré comme une vérité terrain pour le sujet français "son" contient au final 519 mots ; y compris le mot "son", ajouté manuellemant car il relève d'une autre catégorie dans *Le Dictionnaire électronique des mots* (Dubois et Dubois-Charlier (2010)).

Mesures d'évaluation : Nous avons évalué chaque lexique L généré par *Lexifield* en calculant sa précision, son rappel et sa F-mesure par rapport à la liste de référence (vérité terrain) correspondante *LGT*. Comme la liste de référence pour le lexique en français et ceux extraits du Thésaurus de Roget fournissent la catégorie grammaticale du mot, cette information a été prise en compte au cours de l'évaluation. En ce qui concerne LIWC, le rappel est calculé en effectuant l'intersection entre le lexique généré et la liste de référence.

Modèles de référence : Nous avons comparé le lexique construit avec notre approche avec des lexiques produits par deux méthodes de l'état de l'art (Sensicon et Empath) mais aussi avec le modèle de plongement de mots et la liste de départ. Plus précisément, ces cinq lexiques servant de comparaison sont définis comme suit :

Sensicon : La liste des termes extraits de Sensicon (Tekiroglu et al. (2014)) en faisant correspondre un terme à l'un des trois sens si le score pour ce sens est non nul et supérieur à tous les autres scores (uniquement pour les sujets en anglais).

Empath : La liste des termes construits par la méthode Empath (Fast et al. (2016)) en utilisant les vecteurs fournis par leur API[9](uniquement pour les sujets en anglais).

EmpathTT : La liste de termes généré en appliquant la méthode Empath à notre modèle pré-entrainé de plongement de mots.

Max WE : Les mots ayant une similarité de cosinus supérieure à 0,5 avec les graines sont classés en fonction de cette similarité et les k premiers mots sont conservés.

S : La liste initiale S.

Il convient de noter qu'une version sans plongement de mots a aussi été testé mais les résultats ne sont pas présentés faute de place et parce qu'ils n'étaient pas concluants car les définitions d'un dictionnaire contiennent des mots amenant une forte dérive sans pour autant apporter une garantie d'exhaustivité. Il faudrait donc aussi appliquer un filtrage mais on ne peut plus utiliser le dictionnaire pour ce faire puisqu'il a déjà été employé.

Nous évaluons les résultats produits par les différentes étapes de *Lexifield* pour apprécier notamment l'impact de chacune sur la qualité du lexique produit. Ainsi, dans les tableaux des résultats, $L1$ et $L2$ correspondent respectivement au lexique obtenu après l'expansion par plongement de mots et le filtrage des mots à l'aide d'un dictionnaire. De plus, comme l'étape d'ajout de synonymes peut être répétée, par souci de clarté, nous précisons à chaque fois le nombre d'itérations. Enfin, la liste de synonymes pré-filtrée (*i.e.* $L3a$) est indicée par "a" par

9. https://github.com/Ejhfast/empath-client

	Précision	*Rappel*	*F-mesure*	Taille
S	**100**	9,8	17,9	7
Sensicon	1.5	**78 ,8**	2,9	5976
Max WE (k=170)	35,1	29,5	32	108
EmpathTT (k=50)	**44,6**	19,6	27,2	47
Empath (k=150)	33	**32,3**	**32,7**	124
L1	24,7	43,6	31,6	319
L2	**49,2**	29,5	36,9	71
L3a-cycle 1	32,9	47,8	39	176
L3b-cycle 1	36,3	43,3	39,6	143
L3a-cycle 2	30,5	**50,6**	38	200
L3b-cycle 2	34,3	47,8	**39,9**	166

TAB. 1 – *Résultats pour le sujet aggrégé "odour ∪ taste" (LIWC)*

opposition à "b", le liste finale réduite avec dictionnaire (*i.e.* $L3b$). Séparer ces résultats permet de mesurer l'impact du filtrage par dictionnaire sur la performance globale du système.

4 Résultats

4.1 Évaluation du lexique pour ("odour" ∪ "taste")

La table 1 présente les résultats pour le sujet ("odour" ∪ "taste") selon la vérité terrain LIWC. Clairement, notre méthode surpasse tous les autres systèmes avec une F-mesure égale à 39,9 alors que Empath obtient un score égal à 32,7, Max WE 32 et les autres modèles de référence ou Sensicon des scores inférieurs. Cependant, ce dernier système a un rappel plus important égal à 78,8 comparé à 47,8 pour *Lexifield*, mais avec une très faible précision (1,5). Cela est dû à la grande taille du lexique produit par Sensicon (5976) par rapport à la taille de la vérité terrain LIWC qui est limitée à 71 mots. Le même comportement peut être observé avec L1, le lexique obtenu après la première étape de notre système qui vise à étendre la liste de graines, menant à une augmentation du rappel (43,6) par rapport à la liste des graines (9,8) en raison de l'ajout de nouveaux termes (319 pour L1 contre 7 pour la liste des graines S) mais avec une perte de précision (24,7 pour L1 contre 100 pour S car toutes les graines appartiennent à la vérité terrain). Cela souligne l'impact de la taille du lexique sur les scores de précision/rappel et un avantage de notre méthode est que l'utilisateur peut accorder la priorité à une de ces mesures en ajustant la taille de L1, une taille plus grande donnant un meilleur rappel et une taille inférieure induisant une plus grande précision.

4.2 Évaluation pour le sujet "son"

Les résultats obtenus pour le lexique français sont présentés dans la table 2. Notre méthode surpasse les modèles de référence avec une F-mesure égale à 35,6, deux fois supérieure à la F-mesure du lexique généré avec la méthode EmpathTT qui obtient 17 et à Max WE qui atteint 14,3. De plus, cette amélioration concerne la précision (37,6) ainsi que le rappel (33,9) pour

	Précision	*Rappel*	*F-mesure*	Taille
S	**100**	0,5	1,1	3
Max WE	15,4	13,4	14,3	454
EmpathTT	18,0	**16,8**	**17,0**	466
L1	5,6	30,6	9,5	2810
L2	**41,8**	20,8	27,7	258
L3a cycle 1	38,6	33,1	**35,6**	445
L3b cycle 1	40,6	31	35,1	396
L3a cycle 2	33,3	**36,8**	35	572
L3b cycle 2	37,6	33,9	**35,6**	467

TAB. 2 – *Évaluation des modèles de référence et de Lexifield sur le lexique français pour le sujet "son"*

un lexique ayant environ la même taille (467) que ceux produits par les modèles de référence (454 pour Max WE et 466 pour EmpathTT). Comme remarqué précédemment, la précision est maximale, égale à 100 % pour la liste des graines S puisque celle-ci est réduite à 3 termes appartenant également à la vérité terrain mais, en retour, le rappel de cette liste est très faible (0,5) conduisant également à un score de F-mesure très faible (1,1). Cette conclusion concernant la liste de graines reste la même pour les trois autres sujets étudiés.

On peut aussi voir dans la table 2 (ligne L1) que l'expansion basée sur le plongement de mots augmente la taille du lexique de 3 graines à 2810 mots et par conséquent le rappel, qui atteint 30,6 mais elle introduit beaucoup de bruit comme l'indique le taux de précision de 5,6. Cette liste étendue est filtrée efficacement en utilisant les définitions du dictionnaire (L2) comme indiqué par le taux de précision qui est égal à 41,8 mais au prix d'une diminution du rappel (de 30,6 à 20,8). La dernière étape du processus, l'expansion basée sur les synonymes (ligne L3), permet de conserver approximativement la précision obtenue dans l'étape précédente (de 41,8 à 38,6) tout en ayant un rappel du même ordre que celui atteint en utilisant le plongement de mots (33,1). Enfin, il convient également de noter que la F-mesure reste plus ou moins la même lors de l'ajout et du filtrage de synonymes. Ainsi, une itération de la dernière étape suffit pour obtenir une amélioration significative tant pour la précision que pour le rappel (cycle 1 par rapport au cycle 2).

4.3 Évaluation du lexique pour le sujet "sound"

La deuxième série d'expériences est consacrée au même sujet mais en anglais, "sound". Notre système a été comparé non seulement à EmpathTT et à Max WE, mais également à Sensicon et au au système Empath initial. Comme indiqué dans le tableau 3 (1ère section), les résultats obtenus avec les modèles de référence ne sont pas satisfaisants, variant entre 7,3 et 10,6. Il semble que Sensicon permet de récupérer plus de mots (3972), ce qui induit un meilleur rappel (16,6) alors que la taille du lexique produit par EmpathTT est réduite à 1019 mots conduisant à une meilleure précision (8,4). Les résultats obtenus avec Max WE sont intermédiaires. Enfin, il convient de noter que notre variante améliore la performance d'EmpathTT, puisque la F-mesure est égale à 10,6 contre 8,8 pour Empath.

Sujet “sound”	*Précision*	*Rappel*	$F\text{-}mesure$	Taille
S	**100**	0,5	1	3
Sensicon	2,4	**16,6**	4,3	3972
Max WE	6,1	9	7,3	871
EmpathTT	8,4	14,2	**10,6**	1019
Empath	12,5	6,8	8,8	272
L1	4,4	16,7	7	2260
L2	16,6	14,5	15,5	523
L3a cycle 1	16,5	25,1	19,9	905
L3b cycle 1	**18,4**	23,8	**20,7**	771
L3a cycle 2	13,9	**27,6**	18,5	1179
L3b cycle 2	15,6	26,6	19,7	1014
Sujet “taste”	*Précision*	*Rappel*	$F\text{-}mesure$	Taille
S	**100**	1,6	3,2	6
Sensicon	1,6	**24,1**	3,1	5255
Max WE	9,5	9,5	9,5	367
EmpathTT	6	10,5	7,7	640
Empath	7,5	13,6	**9,7**	568
L1	4,0	24,4	7	2199
L2	14,5	19,8	16,8	501
L3a cycle 1	10,5	33,1	16	1156
L3b cycle 1	17,8	29,8	22,3	615
L3a cycle 2	10,7	**34,5**	16,4	1173
L3b cycle 2	**17,9**	30,9	**22,7**	634
Sujet “odour”	*Précision*	*Rappel*	$F\text{-}mesure$	Taille
S	**100**	3,5	6,8	6
Sensicon	2,8	20	4,9	1214
Max WE	11	23	15	362
EmpathTT	7,3	**25,8**	11,4	600
Empath	10,7	14	12,1	204
L1	2,4	41,7	4,5	2958
L2	**15**	35,8	**21,2**	405
L3a cycle 1	9,2	**43,5**	15,2	799
L3b cycle 1	11,9	39,4	18,3	561
L3a cycle 2	8,6	43,5	14,4	853
L3b cycle 2	11,2	39,4	17,4	597

TAB. 3 – *Évaluation des modèles de référence et de Lexifield sur le lexique anglais pour les sujets “sound (en haut)”, “taste ” (au milieu) et “odour” (en bas)*

Comme le montre la table 3, le résultat fourni par notre système est à nouveau le double du meilleur score des quatre modèles de référence avec une F-mesure égale à 20,7 (comparé à 10,6 pour EmpathTT, le meilleur des modèles de référence), et comme dans l'expérience précédente, le gain concerne à la fois le rappel et la précision pour approximativement la même taille de lexique (1014 mots avec deux itérations de l'étape 3 et 1019 mots pour EmpathTT). Ces résultats confirment notre analyse précédente : l'étape 1 augmente la taille du lexique puisque L1 contient 2260 mots ; le rappel atteint donc 16,7, mais il faut l'étape 2 pour filtrer les mots en augmentant ainsi de façon très significative la précision de 4,4 à 16,6 pour L2. Enfin la dernière étape, basée sur l'expansion par ajout de synonymes, renforce l'amélioration avec une F-mesure égale à 20,7 correspondant à une précision de 17,9 et à un rappel de 23,8. En outre, cette deuxième série d'expériences montre bien que notre système n'est pas dépendant de la langue puisque le même algorithme peut être appliqué pour construire efficacement un lexique sur un même sujet dans différentes langues lorsque les ressources linguistiques sont disponibles.

4.4 Évaluation des lexiques pour les sujets "taste" et "odour"

La table 3 présente aussi les résultats obtenus pour les sujets "taste" et "odour" (respectivement au milieu et en bas de la table). La conclusion est la même, confirmant la robustesse de la méthode proposée et son intérêt. De plus, on peut noter les variations de la F-mesure en fonction du nombre d'itérations réalisées lors de l'expansion basée sur les synonymes. Curieusement, la F-mesure la plus élevée pour le sujet "odour" correspond au lexique avant l'extension à base de synonymes ; ce qui peut s'expliquer par la petite taille de la liste de référence.

5 Conclusion

Dans cet article, nous avons présenté une méthode pour construire un lexique spécifique à un domaine à partir d'une liste de mots clés fournis par l'utilisateur grâce à des ressources lexicales. Contrairement aux systèmes de l'état de l'art, nous n'utilisons qu'une très courte liste de termes de départ. De plus, notre méthode, Lexifield, est entièrement automatique et peut être utilisée sans aucune aide manuelle. Enfin elle est indépendante de la langue à condition que les ressources linguistiques soient disponibles. Les résultats expérimentaux obtenus sur plusieurs sujets ont confirmé ses bonnes performances, meilleures ou équivalentes à celles des méthodes de l'état de l'art, quelle que soit la mesure utilisée (précision, rappel et F-mesure). Cependant, le problème traité s'avère difficile car les valeurs absolues de F-mesure restent faibles ; ce qui devrait donner lieu à des travaux complémentaires pour améliorer encore le système.

Références

Baker, C. F., C. J. Fillmore, et J. B. Lowe (1998). The Berkeley Framenet project. In *Proceedings of the 17th international conference on Computational linguistics*, pp. 86–90.

Bojanowski, P., E. Grave, A. Joulin, et T. Mikolov (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics 5*, 135–146.

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

Dubois, J. et F. Dubois-Charlier (2010). La combinatoire lexico-syntaxique dans le dictionnaire électronique des mots. les termes du domaine de la musique à titre d'illustration. *Langages* (3), 31–56.

Fast, E., B. Chen, et M. S. Bernstein (2016). Empath : Understanding topic signals in large-scale text. In *Conference on Human Factors in Computing Systems*, pp. 4647–4657.

Fellbaum, C. (1998). *WordNet : An electronic lexical database*. Bradford Books.

Globerson, A., G. Chechik, F. Pereira, et N. Tishby (2007). Euclidean embedding of co-occurrence data. *J. Mach. Learn. Res. 8*, 2265–2295.

Jakubíček, M., A. Kilgarriff, V. Kovář, P. Rychlỳ, et V. Suchomel (2013). The tenten corpus family. In *7th International Corpus Linguistics Conference CL*, pp. 125–127.

Lavelli, A., F. Sebastiani, et R. Zanoli (2004). Distributional term representations : An experimental comparison. In *CIKM*, pp. 615–624.

Levy, O. et Y. Goldberg (2014). Neural word embedding as implicit matrix factorization. In *NIPS*, pp. 2177–2185.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, et J. Dean (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119.

Park, D., S. Kim, J. Lee, J. Choo, N. Diakopoulos, et N. Elmqvist (2018). Conceptvector : text visual analytics via interactive lexicon building using word embedding. *IEEE Transactions on Visualization & Computer Graphics* (1), 361–370.

Pennebaker, J. W., M. E. Francis, et R. J. Booth (2001). Linguistic inquiry and word count : LIWC 2001.

Riloff, E. et J. Shepherd (1997). A corpus-based approach for building semantic lexicons. In *EMNLP*, pp. 117–124.

Roark, B. et E. Charniak (1998). Noun-phrase co-occurrence statistics for semiautomatic semantic lexicon construction. In *International Conference on Computational Linguistics*, pp. 1110–1116.

Tekiroglu, S. S., G. Özbal, et C. Strapparava (2014). Sensicon : An automatically constructed sensorial lexicon. In *EMNLP*, pp. 1511–1521.

Summary

We present a fully automatic language-independent system for building domain-specific lexicons from a short list of terms defining the domain. *Lexifield* relies on a pre-trained word embedding model, a definition dictionary and a dictionary of synonyms. As compared to other word embedding based systems and a state-of-the-art sensorial lexicon, our system achieves better precision and recall on reference lists extracted from manually created resources such as Roget's Thesaurus.