Réseau Sémantique à partir d'un dictionnaire

Bawden Rachel, Brogniez Caroline et Parslow Nicholas

1 Introduction

2 La Conception du Graphe

La possibilité de construire un réseau sémantique à partir d'un dictionnaire vient du fait que la structure des dictionnaires permet de faire apparaîter des liens sémantiques particuliers. La structure en sens, sous-sens, définitions, exemples etc., mais aussi la structure interne des définitions contient une régularité qu'il est important d'exploiter au maximum. C'est pour cette raison que nous tenons à conserver le plus possible cette structure dans la transformation de dictionnaire en graphe. Un graphe est défini formellement comme étant un ensemble de sommets et un ensemble d'arcs qui relient une paire de sommets.

$$G =< S, A >$$

Pour représenter un dictionnaire par un graphe, nous considérons que les sommets peuvent être les mots individuels du dictionnaire ou même les niveaux intermédiaires de la structure tels que 'exemple', 'définition', 'synonyme', 'antonyme' etc. Les arcs sont alors les liens qui lient les différents éléments d'une entrée de dictionnaire et permettraient de trouver un lien entre un lexème donné et la manière dont il est décrit dans son entrée du dictionnaire.

2.1 Remarques sur le vocabulaire

Nous appelons 'mot' toute unité minimale du lexique. Un mot peut être soit fléchi, soit non-fléchi et par défaut nous faisons référence aux mots non-fléchis sous leur forme de dictionnaire. Par principe, nous restreignons le réseau aux lemmes, mais il est possible qu'il y apparaît des formes fléchies en cas de non-identification du lemme.

Par 'entrée' de dictionnaire nous faisons référence à un groupe d'informations (catégories syntaxiques, sens, définitions, exemples etc.) associées à un mot

donné. Par conséquent, le terme 'entrée' peut aussi être utilisé pour dénoter le mot lui-même, et par extension les informations contenues pour ce mot donné.

La relation sémantique de synonymie est définie entre deux termes de la même catégorie de discours qui ont le même sens et qui peuvent donc être substitués l'un pour l'autre sans modifier le sens de la phrase. Cette définition pose évidemment des problèmes, surtout à cause du fait qu'il est toujours possible de trouver une différence de sens ou d'usage entre deux mots malgré le fait qu'ils soient habituellement classés en synonymes. C'est pour cette raison que parfois il est souhaitable de parler de proche-synonymes au lieu de synonymes tout court. Néanmois, nous préférons utiliser le terme 'synonyme' pour parler de ces cas, sans postuler de théorie sur les frontières de la synonymie. Par la suite, la synonymie sera définie en termes de relations attestées dans des ressources externes et nous nous reportons à ces références pour établir si deux mots sont en relation de synonyme ou pas.

De même pour les relations d'antonymie, d'hyperonymie et d'hyponymie. L'antonymie est définie comme la relation entre deux mots à sens opposé. L'hyperonymie entre un mot dont l'extension contient l'extension d'un autre mot (par exemple, 'véhicule' est l'hypernym de 'voiture'). l'hyponymie est la relation inverse d'hyperonymie, entre un mot dont l'extension est incluse dans l'extension d'un autre (pour reprendre le même exemple, 'voiture' est un hyponyme de 'véhicule').
[AUTRES DEFINITIIONS]

3 La structure des dictionnaires utilisés

Les deux ressources qui seront utilisées sont le Wiktionnaire français en format XML fourni par M. Franck Sajous [REF] et le Littré , qui est aussi disponible en format XML [REF].

Les informations contenues dans les deux dictionnaires sont similaires. Chaque dictionnaire est organisé en entrées, et chaque entrée contient plusieurs définitions, des exemples, des synonymes et des informations grammaticales. Le wiktionnaire contient en plus des relations sémantiques telles que l'antonymie, l'hyperonymie et l'hyponymie.

3.1 Statistiques

NOMBRE d'ENTREES NOMBRE DE CATS SYNT DIFF NOMBRE DE MOT-FORMES DIFF

4 Pré-traitement

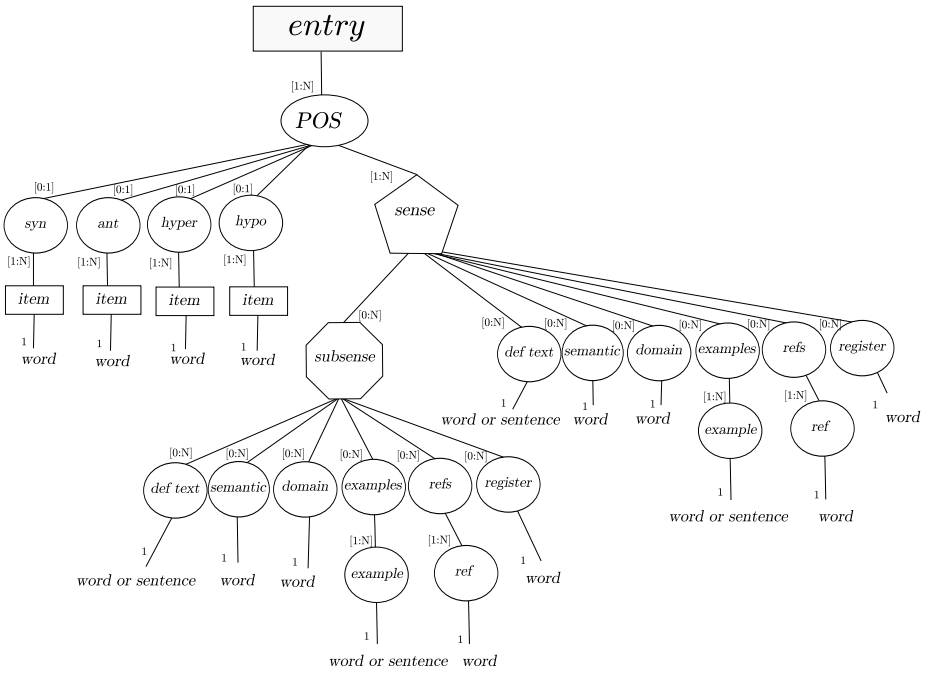
Dans le but de pouvoir procéder à un traitement homogène des deux dictionnaires, une normalisation des deux formats était nécessaire. Ce pré-traitement a permis à la fois de supprimer certaines informations qui n'étaient pas utilisée par la suite et de convertir les balises et leur contenu en un format plus convenable pour notre traitement. Nous réunissons les deux structures dans un seul type de fichier XML avec les balises suivantes :

- entry : un mot du dictionnaire (lexème)
- pos : la catégorie syntaxique (il peut y en avoir plusieurs par lexème)
- sense : le niveau hiérarchique représentant plusieurs sens d'un même mot
- subsense : le niveau hiérarchique représentant plusieurs sous-sens d'un même sens global
- def-text : le texte d'une défintion
- semantic : l'emploi sémantique d'un mot (figuratif, absolu etc.)
- register : le registre du mot (familier, vulgaire, soutenu etc.)
- domain : le domaine sémantique du mot (géographie, biologie, culinaire etc.)
- refs : des références vers d'autres entrées lexicales
- examples : des phrases exemples contenant le mot de l'entrée lexicale
- synonyms : synonymes de du mot de l'entrée
- antonyms : antonymes de du mot de l'entrée
- hyperonyms : hyperymes de du mot de l'entrée
- hyponyms : hyponymes de du mot de l'entrée

La DTD des fichiers XML normalisés se trouve dans l'annexe [REF], avec les scripts de normalisation [REF]. La structure suivante illustre l'hiérarchie des balises, qui correspond aussi à la structure du réseau :

De légères différences existent entre les structures des deux dictionnaires. Tandis que la plupart des informations du Wiktionnaire sont localisées au niveau du 'sense', la plupart des informations du Littré sont localisées au niveau du 'subsense', ce qui ne pose aucun problème pour la construction des deux réseaux.

En plus de la restructuration des fichiers XML, les dictionnaires ont été nettoyés pour convenir à nos besoins. Certaines balises contiennent trop d'informations ou des informations non-pertinentes. Par exemple, les balises des commentaires pour le Littré ne contiennent pas de contenu exploitable et les citations des textes anciens des mots vieillis qui risquent d'ajouter du bruit dans le réseau résultant. Les entrées sont ainsi réduites au schéma ci-dessus [FIG]. Le Littré en particulier nécessite une étape importante de nettoyage, puisque des erreurs existent dans le balisage des données et un manque d'homogénéité dans la structuration des entrées risque de perturber la manière dont laquelle les relations sont établies entre les éléments qu'elles contiennent. Les catégories syntaxiques utilisées ne sont pas d'une liste énumérable et contiennent des descriptions plutôt



littéraires. Cette étape de normalisation consistait aussi à traduire ces catégories syntaxiques en une liste plus formelle et identifiable.

Les définitions, exemples et autres informations dans les entrées étaient ensuite taggés et lemmatisés en utilisant l'outil MElt [REF] et le tokeniseur Bonsai [REF]. Cette étape est importante pour pouvoir relier les mots fléchis des définitions avec leurs lemmes tels qu'ils apparaissent dans les entrées et pour savoir de quelle catégorie syntaxique il s'agit. Le script XXX qui sert à tagger et lemmatiser les documents se trouvent dans l'annexe [FIG].

5 La représentation en graphe A REFAIRE – plus clair

Les fichiers XML sont facilement transférables en représentation en graphe, puisqu'ils continennent une hiérarchie d'entrées. En principe chaque niveau de l'hiérarchie est représenté par un nœud différent avec des arcs qui lie chaque nœud à ses fils dans l'hiérarchie, comme dans [REF SCHEMA]. Cette représentation a l'avantage de préserver la structure hiérarchique d'origine.

Les arcs entre nœuds sont orientés, mais nous créeons aussi des liens inverses séparées pour pouvoir remonter dans chaque niveau de l'hiérarchie. Pour in-

jecter plus de sophistication dans le réseau, la distance entre deux mots ne se limite pas au nombre d'arêtes dans le chemin. Les arêtes sont pondérées selon le type de nœud sortant et entrant afin de distinguer entre les relations différentes qui peuvent exister à différents endroits de l'entrée. Les poids des arêtes font partie des paramètres du réseau et sont conservés dans un fichier de configuration externe (weight.config), les valeurs étant choisies pour optimiser le réseau pour une tâche particulière. L'optimisation sera discutée dans la partie [XXX]. Il existe un total de 42 liens différent [CHECK] avec un poids différent entre zéro et infinité, un lien de 0 étant un lien non-existant dans le graphe. Ci-dessus un extrait de ce fichier, qui se trouve dans l'annexe [REF] :

```
entry2pos = 0.01
pos2entry = 0.01
pos2sense = 0.01
sense2pos = 0.01
pos2syn = 0.01
syn2pos = 0.01
...
sense2subsense = 0.001
subsense2sense = 0.001
subsense2deftext = 0.00001
deftext2subsense = 0.00001
subsense2semantic = 0
```

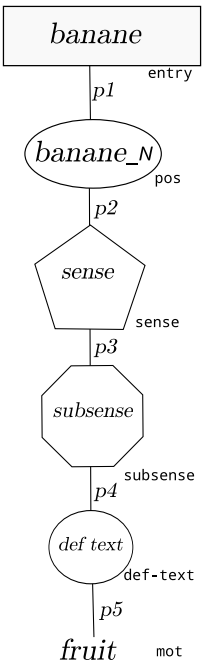
6 La création du graphe

6.1 Choix de langage et de bibliothèques

Nous implémentons le graphe en python, pour lequel il existe de nombreuses bibliothèques (Scipy, NumPy) efficaces qui permettent de manipuler un grand nombre de données numériques. Nous utilisons cElementTree pour traiter les fichiers XML et les SparseMatrices de NumPy pour représenter le graphe. Le réseau est alors une matrice NxN où N est le nombre de nœuds différents dans le graphe. Etant donné le grand nombre d'entrées dans les dictionnaires, il y a un avantage clair d'utiliser les matrices sparses, qui permettent de stocker plus efficacement un graphe qui contient de nombreux sommets et peu d'arêtes.

6.2 Simplification de la structure

En pratique, il est possible de surpasser d'un grand nombre de nœuds intermédiaires dans l'hiérarchie en attribuant une arête directe entre une paire



de mots dont le poids serait l'addition de toutes les valeurs des liens qui constituent le chemin entre les deux mots. Le choix des sommets est un compromis entre mettre le plus d'informations possible dans le graphe et veiller à la non-explosion de la taille du graphe. Ceci n'est que possible parce que dans les tâches effectuées (détaillés dans la partie XXX), nous n'aurons jamais besoin d'extraire ces nœuds intermédiaires, même si nous souhaitons tenir compte de leur présence.

Ainsi, l'entrée dans la figure XXX est simplifiée en l'entrée dans la figure XXX+1 :

La structure de graphe décrit précédemment ne détaille pas quels liens vont exister et entre quels nœuds. Pour représenter la structure hiérarchique complète du dictionnaire, il pourrait exister pour chaque niveau de la hiérarchie (POS, sense, subsense, deftext, example, word etc.) un sommet qui est lié aux autres nœuds dans la hiérarchie. Les arêtes seraient orientées du sommet de l'entrée jusqu'aux mots individuels en bas de la hiérarchie, afin de créer un chemin entre le lexème d'une entrée et les mots qui appartiennent à sa description.

Le choix des sommets est un compromis entre mettre le plus d'informations possible dans le graphe et veiller à la non-explosion de la taille du graphe. Si une entrée contient des informations à chaque niveau de la structure (un synonyme, un antonyme, une définition, un exemple etc.), pour cette seule entrée, le nombre de nœuds intermédiaires ajoutés est de vingt-trois. NOMBRE D'ENTREES TOTAL. Eliminer totalement ces nœuds intermédiaires (ceux qui ne correspondent pas aux mots individuels du dictionnaire) risque d'aplatir la structure du dic-

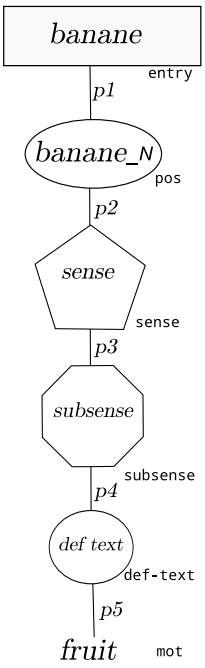


FIGURE 1 – First.

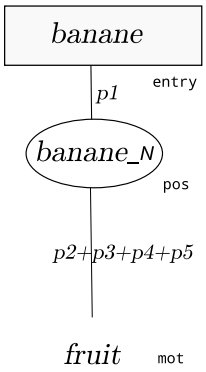


FIGURE 2 – Second.

tionnaire est de perdre l'avantage hiérarchie du départ.

Les différents relations à l'intérieur de l'entrée sont mise en évidence par le fait que les arêtes entre les différentes informations soient pondérées, avec un poids spécifique pour chaque paire de nœuds sortant et entrant. [EXPLIQUE PLUS CLAIREMENT.]

Ayant pour but de représenter le moins de nœuds intermédiaires que possible dans le graphe lui-même pour diminuer le temps de parcours, en conservant la structure hiérarchique, nous utilisons les poids de ces arêtes différentes pour initialiser les poids des arêtes dans le graphe, sans devoir passer par certains nœuds lors d'un parcours.

Par exemple, il est possible d'éliminer le nœud 'subsense', s'il existe une arête de 'sense' à chaque fils de 'subsense' qui a un poids qui est l'addition du poids pour l'arête spécifique 'subsense' à 'fils' et du poids 'sense' à 'subsense'.

[PLUS]

En plus des arêtes descendantes qui existent dans le schéma [FIG], nous établissons des arêtes montantes, afin de retrouver facilement la relation entre un mot qui apparaît dans une entrée et le lexème de l'entrée. Les poids sur ces arêtes ne sont pas les mêmes que les arêtes descendantes afin de retrouver une différence dans ces relations.

[IMAGE plein d'arêtes].

6.2.1 Restructuration des fichiers XML

Bien qu'en format XML, la structure du littré ne respecte pas la DTD fournit et un pré-traitement conséquent s'est avéré nécessaire avant son exploitation. Parmi les étapes de pré-traitement étaient :

- le stockage des alternatifs orthographiques dans un dictionnaire pour utilisation ultérieur
- la suppression d'alternatifs flexionnels (ex : la forme "emboué -ée" est réduite à "embouée")
- la normalisation des catégories syntaxiques, originellement en écrit en prose, vers un ensemble fini de catégories syntaxiques possible (ex : "s. f. et mieux s. m" devient simplement "N" pour 'nom')
- la suppression de certaines balises (commentaires, vieilles citations, prononciation) qui pourraient introduire du bruit

Le résultat est un unique fichier XML dans lequel les informations associées aux entrées sont toutes contenues à l'intérieur de la balise 'subsense'. Cette structure hiérarchique correspond à la structure originelle du Littré dans lequel les items lexicaux sont organisés d'abord par la forme du mot et ensuite par la catégorie syntaxique. Plusieurs sens différents peuvent être trouvés lorsqu'un mot d'un certain catégorie ne se trouve pas dans la même entrée que son entrée principale dans la version originelle du Littré. Le schéma suivant démontre la structure du fichier résultant :

6.2.2 Le fichier XMl produit