

# Si beaucoup d'oiseaux volent, peut-on en déduire que tous les oiseaux volent ?

## Production endogène de règles déductives dans un réseau lexico-sémantique

Mathieu Lafourcade<sup>1</sup> Alain Joubert<sup>1</sup>

(1) LIRMM, 161 Rue Ada, 34095 Montpellier, France

Mathieu.Lafourcade@lirmm.fr, Alain.Joubert@lirmm.fr

### RÉSUMÉ

---

A partir d'un réseau lexico-sémantique, il est possible de générer des règles de façon inductive à partir des faits présents. Ces règles permettent de densifier le réseau et d'en réduire les silences. Afin de minimiser l'émergence de relations qui pourraient être erronées, la question de la polysémie est abordée et un filtrage sur les règles présentant des exceptions est réalisé.

### ABSTRACT

---

#### Endogenous production of deductive rules in a lexical-semantic network

With a lexical-semantic network, it is possible to discover rule inductively from known facts. These rules are instrumental in reducing missing information in the network. In order to minimize emergence of relations that could be erroneous, we approached the question of polysemy and a filtering on rules presenting exceptions is realized.

MOTS-CLÉS : découverte de règles, inférence, base de connaissances, réseau lexico-sémantique.

KEYWORDS: rule discovery, inference, knowledge base, lexical-semantic network.

---

## 1 Introduction

Le réseau lexico-sémantique JeuxDeMots (JDM) est constitué de termes (mots ou expressions) reliés par des relations sémantiques typées et pondérées. Il est initialement renseigné par des joueurs grâce à des jeux sérieux (ou GWAP), ainsi que par des contributeurs qui interviennent directement, tel que décrit dans (Lafourcade *et al.*, 2015). En outre, le réseau est densifié par des mécanismes d'inférence automatique endogène de relations reposant sur des processus de déduction, d'induction ou d'abduction (Zarrouk et Lafourcade, 2014); il est également densifié de manière exogène par extraction de relations depuis des ressources externes, telles que Wikipédia ou des corpus littéraires ou journalistiques (Lafourcade et Joubert, 2013).

Un réseau lexical est composé de noeuds (qui habituellement sont des termes du lexique, ou des concepts) ainsi que de relations sémantiques. Ces relations peuvent être pondérées et éventuellement annotées. La notion de règle n'est pas présente dans un tel réseau mais on peut y trouver un intérêt. Ce qu'on entend usuellement par *règle* sont un certain nombre de conditions à vérifier sur un objet associé à une transformation sur ce même objet, transformation qui sera effectuée si les conditions sont vérifiées. En plus de pouvoir densifier une ressource, une règle peut être utilisée comme forme

de précompilation de connaissances utilisable en analyse sémantique de texte lorsqu'on est confronté à un mot inconnu, pour lequel on ne dispose que d'informations contextuelles (*XYZ est un nouvel antipyrétique*).

Dans cet article, nous montrons qu'il est possible de densifier le réseau grâce à un mécanisme de règles, découlant d'un certain nombre d'observations de la vie courante (par ex : *un oiseau vole*) et appliquées sur les relations présentes dans le réseau. Dans une première partie, nous définirons la notion de règle, ainsi que les avantages qui en découlent, puis nous verrons comment celles-ci peuvent être produites, évaluées et utilisées. Nous aborderons la question de la polysémie. Nous montrerons ensuite les résultats obtenus sur le réseau JeuxDeMots (JDM) qui nous permettront de conclure sur l'efficacité de ce processus de découverte de règles.

## 2 Notion de règle

Un certain nombre de travaux ont été effectués sur l'induction de règles de raisonnement, en particulier depuis ceux de (Genesereth et Nilsson, 1987). Citons par exemple ceux de (Berant et al., 2012) et (Berant et al., 2015) qui proposent des méthodes automatiques pour la détection de relations sémantiques et la construction de taxonomies.

Nous nous focalisons dans ce qui suit sur la notion de règle dans le cadre d'un réseau lexico-sémantique. Dans un tel réseau, les relations entre termes sont typées (avec un ensemble représentatif de types sémantiques), et portent des pondérations éventuellement négatives.

### 2.1 Définition des règles

De façon abstraite, une règle est un ensemble non nul de prémisses suivi d'un ensemble non nul de conclusions.

$$\{\$x \text{ relation terme}\}_+ \Rightarrow \{\$x \text{ relation terme}\}_+$$

Si un terme  $\$x$  du réseau vérifie toutes les prémisses alors les relations en conclusion seront ajoutées à  $\$x$ . Il existe déjà un certain nombre de règles, relevant du sens commun, définies manuellement par des contributeurs pour le réseau JDM. Par exemple, *un oiseau vole*, ou encore *un mammifère est vivipare* qui s'écrivent formellement :

$$\$x \text{ r\_isa oiseau} \Rightarrow \$x \text{ r\_agent-1 voler}$$

$$\$x \text{ r\_isa mammifère} \Rightarrow \$x \text{ r\_carac vivipare}$$

$$\$x \text{ r\_isa cétacé} \wedge \$x \text{ r\_has\_part fanons} \Rightarrow \$x \text{ r\_isa mysticète}$$

$$\$x \text{ r\_has\_part boutons} \wedge \$x \text{ r\_has\_part manches} \Rightarrow \$x \text{ r\_isa vêtement}$$

(où  $\$x$  représente un terme quelconque du réseau JDM). Ces règles sont généralement vraies, mais elles peuvent toutefois présenter des exceptions : *une autruche ne vole pas*, *un ornithorynque n'est pas vivipare*. Dans JDM, les contributeurs ont ainsi défini un certain nombre de règles permettant de densifier le réseau, comme par exemple celles indiquées ci-dessus.

Considérons une relation inférée à l'aide de la première règle présentée ci-dessus (rappel :  $\$x \text{ r\_isa oiseau} \Rightarrow \$x \text{ r\_agent-1 voler}$ ) :

- si elle existe déjà dans le réseau JDM, éventuellement négative (par exemple : autruche  $r\_agent-1 \text{ voler} < 0$ ), cela n'entraîne aucune conséquence sur le réseau. On suppose que soit l'information est déjà connue, soit il s'agit d'une exception déjà repérée ;
- si la relation inférée est nouvelle ; elle n'est pas directement introduite dans le réseau JDM : elle est soumise à validation (par un expert ou par un système de vote majoritaire auprès de contributeurs), en raison de l'existence éventuelle d'exceptions ou de polysémie.

Tout comme le fait un humain, n'est-il pas possible à partir d'exemples de développer un processus inductif de production automatique de règles ?

Avant de poursuivre, il est sans doute nécessaire de préciser quelques propriétés du réseau lexico-sémantiques JeuxDeMots (JDM). Ce réseau porte sur la langue française et est construit par externalisation ouverte (crowdsourcing) par l'entremise de jeux (des GWAP - Games With A Purpose) (Lafourcade, 2007). Pour ce type d'approche adoptée pour la construction de ressources, on peut renvoyer le lecteur à (von Ahn et Dabbish, 2008), à (Mihalcea et Chklovski, 2003), à (Zesch et Gurevych, 2009) à (Lieberman *et al.*, 2007), mais aussi (Marchetti *et al.* 2007). Concernant le type d'information contenu, il est dans l'esprit de Wordnet (Miller *et al.* 1990), de Eurowordnet (Vossen, 1998), du Wolff (Sagot et Fier, 2008), de Babelnet (Navigli et Ponzetto, 2012), de HowNet (Dong et Dong, 2006) mais également de Cyc (Lenat, 1995). Toutefois un certain nombre de différences méritent d'être mentionnées :

- les termes polysémiques sont associés à leur différents sens (on parlera plutôt de raffinements) via une relation de raffinement vers des termes où le terme est glosé. Par exemple, le terme *frégate* est associé à deux raffinements *frégate>navire* et *frégate>oiseau*. L'opérateur > indique une glose associée au terme polysémique. L'entrée du raffinement a comme étiquette la séquence : *terme>glose*.
- les relations entre termes sont typées et surtout pondérées. Un poids représente la force d'association entre les deux termes via-à-vis du type de relations. Par exemple, la relation *chat r\_isa : p<sub>1</sub> félin* a un poids  $p_1$  différent du poids  $p_2$  de *chat r\_isa : p<sub>2</sub> animal* ( $p_1$  est sans doute supérieur à  $p_2$ ). Les poids sont obtenus via les jeux qui permettent la construction du réseau.
- des relations à poids négatifs sont introduites dans le réseau pour représenter des informations fausses. Par exemple, *autruche r\_agent-1 : p voler* avec  $p < 0$  (une autruche *est l'agent négatif* de voler = une autruche ne peut pas voler).
- JDM contient environ 120 types de relations différents<sup>1</sup>. Ces types peuvent être d'ordre lexical (nature morphologique, lemme, etc.), lexico-sémantique (synonyme, antonyme, etc.), sémantique/ontologique (générique, spécifique, parties, tout, lieux typiques, etc.) et enfin faire références à des rôles sémantiques (agent, patient, instrument, but, bénéficiaire, etc.).
- Enfin, des informations conceptuelles (sous forme de symboles distincts des termes du lexique) peuvent être associées aux entrées. Par exemple, le terme *chat* peut être associé aux symboles *\_INFO-SEM-THING-NATURAL*, *\_INFO-SEM-LIVING-BEING*, respectivement un objet naturel ou un être vivant).

Dans le réseau JDM, toutes les informations associées à un terme le sont via des relations. Les objets du réseau n'ont pas d'attributs ou de propriétés qui seraient représentés d'une autre façon.

1. <http://www.jeuxdemots.org/jdm-about-detail-relations.php>

## 2.2 Intérêt de la densification du réseau

Rappelons tout d'abord que, lors de la densification du réseau, il n'y a pas création de nouveaux termes, mais inférence de nouvelles relations entre des termes déjà existants.

La densification du réseau entraîne une augmentation du temps d'accès aux informations, celui-ci étant proportionnel au logarithme en base 2 du nombre de relations (en tenant compte des index). Il pourrait alors être légitime de se poser la question, lors d'une recherche d'information, de l'application de règles sur un réseau non densifié, l'accès au réseau serait alors plus rapide, plutôt que de rechercher dans un réseau densifié par l'application *a priori* de règles. Or, l'application d'une règle nécessite toujours plusieurs accès dans le réseau (en moyenne une vingtaine de requêtes pour une inférence déductive). Il sera donc plus rapide d'accéder à une information dans le réseau densifié, plutôt que d'accéder au réseau non densifié puis d'appliquer des règles d'inférence.

De plus, en raison des exceptions, l'application *a priori* de règles permet une vérification, également *a priori*, de la validité des relations ainsi inférées, ce que ne permet pas une application de règles lors d'une recherche d'informations.

## 2.3 Intérêt de l'application des règles sur les nouveaux termes

L'application des règles a un intérêt tout particulier lors de la détection de nouveaux termes à la faveur d'une analyse de textes, ou lorsqu'ils apparaissent suite à l'activité des joueurs sur JDM. Par exemple, si l'analyse d'un texte permet d'identifier un nouveau terme correspondant à une espèce d'oiseau non encore présente dans le réseau, ce terme sera introduit dans le réseau, et par application des règles portant sur les oiseaux, il "héritera" de toutes leurs caractéristiques. Il faudra toutefois vérifier que cette espèce d'oiseau ne présente pas des exceptions par rapport à l'ensemble des oiseaux en général, comme par exemple la non capacité à voler.

## 2.4 Production automatique de règles

Si dans le réseau, nous avons plusieurs termes  $a$  tels que :

$$a \text{ } r\_isa \text{ } b \text{ et } a \text{ } R \text{ } c$$

où  $R$  représente un type de relation sémantique, alors nous avons potentiellement la règle : quel que soit le terme  $x$  tel que  $x \text{ } r\_isa \text{ } b$ , alors  $x \text{ } R \text{ } c$ .

Ici, la règle produite s'appuie sur une relation de généralité  $x \text{ } r\_isa \text{ } b$ . Toutefois, ce n'est pas nécessairement toujours le cas. Par exemple, nous avons la règle :

$$\text{si } x \text{ } r\_agent-1 \text{ voler, alors } x \text{ } r\_has\_part \text{ ailes.}$$

Nous pouvons produire des règles potentielles en explorant automatiquement le réseau lexical. Considérons un terme  $a$  du réseau, nous pouvons énumérer l'ensemble des relations sémantiques qu'il entretient avec d'autres termes. Pour chacune de ces relations, que l'on peut noter  $a \text{ } R \text{ } b$ , nous pouvons énumérer tous les termes  $x$  ayant la même relation. Enfin, pour chacun de ces termes, compter le nombre d'occurrences de relations sémantiques (séparément les positives et les négatives).

Par exemple, nous pouvons obtenir les règles suivantes (règle suivie du nombre de termes la vérifiant

et le pourcentage par rapport au nombre de termes vérifiant la prémisse) :

(On rappelle que la notation  $x > y$  correspond à un raffinement du terme  $x$  accompagné de la glose  $y$ . Par exemple, *chat > félin* est le raffinement du terme *chat* glosé par le terme *félin* correspondant au sens correspondant de *chat*.)

- 1 :  $\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_agent-1} : décoller > prendre l'air > aéronautique ; 1767 ; 100\%$
  - 2 :  ~~$\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_isa} : avion > véhicule ; 1765 ; 100\%$~~
  - 3 :  ~~$\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_isa} : avion ; 1764 ; 99.94\%$~~
  - 4 :  $\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_isa} : aéronef ; 1377 ; 78.02\%$
  - 5 :  $\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_agent-1} : atterrir ; 1377 ; 78.02\%$
  - 6 :  $\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_lieu} : ciel ; 1377 ; 78.02\%$
  - 7 :  $\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_agent-1} : voler ; 1377 ; 78.02\%$
  - 8 :  $\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_agent-1} : se poser ; 1377 ; 78.02\%$
  - 9 :  $\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_agent-1} : virer ; 1377 ; 78.02\%$
  - 10 :  $\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_agent-1} : décoller ; 1377 ; 78.02\%$
  - 11 :  $\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_agent-1} : voler > déplacement aérien ; 1377 ; 78.02\%$
  - 12 :  $\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_agent-1} : descendre > aller en bas ; 1377 ; 78.02\%$
  - 13 :  $\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_agent-1} : décoller > prendre l'air ; 1377 ; 78.02\%$
  - 14 :  $\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_agent-1} : se crasher ; 1377 ; 78.02\%$
  - 15 :  $\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_lieu} : air ; 1377 ; 78.02\%$
  - 16 :  $\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_agent-1} : s'écraser ; 1377 ; 78.02\%$
  - 17 :  $\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_has\_part} : train d'atterrissage ; 1377 ; 78.02\%$
  - 18 :  $\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_has\_part} : siège ; 1377 ; 78.02\%$
  - 19 :  $\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_isa} : véhicule ; 1377 ; 78.02\%$
  - 20 :  $\$x_{r\_isa} : avion > véhicule \Rightarrow \$x_{r\_has\_part} : moteur ; 1376 ; 77.96\%$
- ...

Il semble raisonnable de filtrer les règles 2 et 3. En effet, on ne garde pas de règles tautologiques, ni de règle où la conclusion serait un raffinement d'une prémisse unique (ou à l'inverse, le général). La production de ce type de règles est filtrée en amont (et elles ne sont donc jamais proposées).

Enfin, il est possible de réitérer le processus, en ajoutant à chaque règle une prémisse supplémentaire parmi celles trouvées, jusqu'à ne plus trouver aucun exemple positif. Certains filtrages peuvent constituer une heuristique permettant de réduire la complexité algorithmique du processus (voir plus bas le compte-rendu de l'expérience menée).

Par exemple, avec une prémisse de plus :

$\$x_{r\_isa} : avion > véhicule \& \$x_{r\_has\_part} : réacteur \Rightarrow \$x_{r\_has\_part} : entrée d'air ; 73 ; 10.86\%$

...

Il est possible d'explorer le réseau (afin de produire des règles) en tirant aléatoirement un terme et en sélectionnant ses relations sémantiques comme prémisses de départ. Une heuristique un peu plus efficace est de choisir en premier lieu les termes les plus sollicités ou ceux ayant le plus de relations entrantes (les *hubs*).

Comme on peut le constater l'approche est conceptuellement simple, bien que calculatoirement

explosive. En pratique, le travail de fouille de règles étant réalisé hors-ligne, l'approche est réalisable.

## 2.5 Combien d'exemples sont nécessaires pour créer une règle ?

S'il y a beaucoup d'exemples, la règle a probablement déjà été définie par un expert ou trouvée automatiquement et validée. Même si cette règle n'a pas déjà été définie, elle sera probablement peu productive, un grand nombre de relations existant déjà. S'il y a peu d'exemples :

- soit la règle est vraie, et alors elle est potentiellement productive,
- soit la règle traduit des exceptions, et alors elle n'est qu'exceptionnellement vraie.

## 3 Problème de la polysémie

En reprenant les notations précédentes, le terme  $b$ , but de la relation constituant la prémisse d'une règle, peut être polysémique, avec les raffinements  $\mathbf{b>b1}$  et  $\mathbf{b>b2}$ . Dans ce cas, ce terme  $\mathbf{b}$  peut être but de plusieurs relations de même type dont les termes sources n'ont rien d'autre en commun. A savoir les deux règles :

$$\$x \ r\_isa \ b>b1 \text{ et } \$x \ r\_isa \ b>b2$$

en l'absence de relation entre  $b>b1$  et  $b>b2$ , nous conduira probablement à obtenir deux groupes distincts de termes  $\$x$ , groupes sans véritable relation.

Par exemple, si nous avons la règle suivante :

$$\$x \ r\_isa \ maison \Rightarrow \$x \ r\_has\_part \ mur$$

Alors, nous pourrions en déduire que :

$$Larousse \ r\_isa \ maison \Rightarrow Larousse \ r\_has\_part \ mur$$

En fait, le terme *maison* est polysémique : *maison>habitation* et *maison>édition*. La règle ne s'applique que sur le raffinement de  $b$  qui vérifie la deuxième partie. C'est donc ici :

$$\$x \ r\_isa \ maison>habitation \Rightarrow \$x \ r\_has\_part \ mur$$

De manière analogue, si c'est le terme  $a$ , source de la prémisse, qui est polysémique, la règle ne s'applique qu'au raffinement de  $a$  en relation avec le terme  $b$  sur lequel la règle repose.

Par exemple, si nous avons la règle :

$$\$x \ r\_isa \ oiseau \Rightarrow \$x \ r\_agent-1 \ voler$$

Alors nous avons :

$$rapace \ r\_isa \ oiseau \Rightarrow rapace \ r\_agent-1 \ voler.$$

Or *rapace* est polysémique : *rapace>oiseau* et *rapace>personne avide*.

Dans ce cas-là, la règle ne s'applique qu'au raffinement *rapace>oiseau*, même si ici, anecdotiquement, *rapace>personne avide r\_agent-1 voler* est valable pour le raffinement *voler>dérober*.

## 4 Expérience et évaluation

Nous rappelons que nous nous plaçons en monde ouvert, c'est-à-dire que le réseau lexical est considéré comme incomplet et comporte éventuellement des erreurs (ayant vocation à être corrigées). Au moment de l'expérience de génération de règles, le réseau contenait environ 220 millions de relations et 2 millions de termes.

Pour une règle  $R$ , soit un ensemble de prémisses  $\mathcal{P}$ , nous notons alors  $\mathcal{P}_+$  l'ensemble des termes vérifiant ces prémisses. Pour un ensemble de conclusions  $\mathcal{C}$ , nous notons  $\mathcal{C} \setminus \mathcal{P}_+$  l'ensemble des termes de  $\mathcal{P}_+$  vérifiant positivement  $\mathcal{C}$ . De façon similaire, nous notons  $\mathcal{C} \setminus \mathcal{P}_-$  et  $\mathcal{C} \setminus \mathcal{P}_0$  l'ensemble des termes de  $\mathcal{P}_+$  respectivement vérifiant négativement et ne vérifiant pas  $\mathcal{C}$ .

Le confiance d'une règle est le rapport :

$$C_R = \frac{|\mathcal{C} \setminus \mathcal{P}_+|}{|\mathcal{P}_+|}$$

Plus la confiance est proche de 1, plus la règle semble juste, à savoir qu'il existe pas ou peu de termes de  $\mathcal{P}_+$  pour lesquels la conclusion n'est pas positive.

La défiance d'une règle est le rapport des exemples négatifs sur l'ensemble de ceux vérifiant les prémisses :

$$D_R = \frac{|\mathcal{C} \setminus \mathcal{P}_-|}{|\mathcal{P}_+|}$$

Le silence d'une règle est le rapport :

$$S_R = \frac{|\mathcal{C} \setminus \mathcal{P}_0|}{|\mathcal{P}_+|}$$

La question qui se pose maintenant est d'évaluer l'impact respectif des contre-exemples par rapport aux exemples, mais également dans quelle mesure les silences doivent être considérés.

L'approche la plus simple, si l'on désire maximiser le nombre de règles vraies, est d'exclure celles présentant des contre-exemples ( $D_R = 0$ ). Concernant le silence, nous fixons empiriquement qu'on en tolère jusqu'à 80% :  $S_R \leq 0.8$ . Enfin, il faut au minimum 2 exemples ( $|\mathcal{C} \setminus \mathcal{P}_+| \geq 2$ ).

Par ailleurs, nous effectuons un **filtrage sémantique** des règles produites. Une règle  $r$  ne sera pas proposée s'il existe une règle  $R$  dont les prémisses  $\mathcal{P}_R$  sont strictement incluses dans celles de  $r$  ( $\mathcal{P}_r$ ) et que leurs conclusions sont identiques. En effet,  $R$  ayant moins de prémisses est *plus forte* que  $r$ .

A titre d'évaluation, nous avons généré un peu plus de 2 millions de relations dont nous avons évalué manuellement environ 0,1% (2165 relations). Lors de la génération, l'ensemble des filtres évoqués ci-dessus ont été appliqués. Sur les 2165 relations évaluées, 1747 (soit environ 80%) ont été considérées comme justes. Pour les 418 relations incorrectes, un contre-exemple a été fourni au réseau lexical et a suffi pour supprimer l'erreur. Par exemple, la règle suivante avait été proposée :

$\$x \ r\_isa \ avion > \text{véhicule} \ \& \ \$x \ r\_has\_part \ réacteur \Rightarrow \ \$x \ r\_isa \ avion \ de \ combat$

qui est clairement incorrect. Indiquer au réseau, que par exemple *AirBus A380* n'est pas un *avion de combat* suffit à ne plus voir cette règle proposée.

La totalité des règles fausses générées relevait d’une généralisation abusive et aucun contre-exemple n’était présent dans le réseau. L’insertion d’un contre-exemple a immédiatement résolu le problème. Les règles sont donc potentiellement des révélateurs de manques dans le réseau.

On pourrait faire varier le seuil de silence afin d’évaluer l’impact sur la génération de règles. Toutefois, ce seuil doit être haut si on souhaite avoir un mécanisme utile (à savoir combler les silences du réseau).

Nous avons mené un expérience complémentaire afin d’évaluer l’effet d’une tolérance aux nombres d’exceptions. Par rapport à l’expérience mentionnée précédemment, nous avons réduit le parcours de graphe afin d’obtenir différents résultats en temps raisonnable. Pour ce même parcours (parfaitement déterministe) du graphe, nous avons retenu les règles avec une tolérance de respectivement au plus 0, 1, 2, 5 et 10 exceptions. De façon évidente, chaque ensemble de règles produites contient l’ensemble précédent, Nous avons évalué manuellement par tirage aléatoire 200 règles distinctes pour chaque ensemble.

<b>tolérance</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>5</b>	<b>10</b>
# règles produites	5674	7865	21834	123901	869187
estimation					
% règles correctes (# règles évaluées)	85% (200)	67% (400)	32% (600)	8% (800)	2% (1000)
estimation					
# règles correctes différentiel	4823	5270 (+447)	6987 (+ 1717)	9912 (+ 2925)	17384 (+ 7472)

On constate donc sans surprise que plus le mécanisme de découverte de règles est tolérant au nombre d’exceptions plus le nombre de règles produites est grand, mais avec un nombre croissant de règles fausses. Les deux expériences ayant été menées à des moments différents, le graphe ayant évolué entre-temps, ce qui peut expliquer sans doute la différence d’évaluation à tolérance 0 (80 % et 85%).

## 5 Conclusion

Les observations de la vie courante contenues dans le réseau lexico-sémantique JeuxDeMots (ex : les oiseaux volent, les avions aussi) nous ont permis d’élaborer un mécanisme de production automatique de règles. C’est un mécanisme visant à densifier le réseau : il n’y a pas création de nouveaux termes, mais inférence de nouvelles relations entre termes déjà existants.

L’expérience a montré que ce mécanisme est particulièrement productif, et que globalement (80% à tolérance 0) les règles produites sont à la fois justes et pertinentes. Dans l’état actuel de nos travaux, la validation des règles reste manuelle tout comme l’insertion de contre-exemples.

Parmi les pistes de travail futures, il serait intéressant d’explorer plus avant l’équivalence logique des règles entre elles afin de tenter de réduire la taille de l’ensemble. De plus, des mécanismes de révisions de règles doivent être envisagée à mesure que des exceptions sont identifiées.

Enfin, concernant les silences dont nous ne tenons peu compte, il est possible de les réduire en appliquant des mécanismes de prédiction de relations. Un tel mécanisme infère si les relations testées par les prémisses pourraient être vraies ou fausses. On peut pour cela solliciter des mécanismes



logiques et statistiques comme dans (Zarrouk et Lafourcade, 2014) ou des classifieurs neuronaux de type apprentissage profond.

## Références

- BERANT, J., DAGAN, I. and GOLDBERG, J. (2012) : Learning Entailment Relations by global Graph Structure Optimization. *Computational Linguistics*, pages 73-111.
- BERANT, J., ALON, N., DAGAN, I. and GOLDBERG, J. (2015) : Efficient global Learning of Entailment Graphs. *Computational Linguistics*, pages 221-263.
- BLANCO, E. and MOLDOVAN, D. (2011). A model for composing semantic relations. Ninth International Conference on Computational Semantics (IWCS'11), Oxford, United Kingdom, pages 45-54.
- DONG, Z. and DONG, Q. (2006). *HowNet and the Computation of Meaning*. WorldScientific, London.
- FELLBAUM, C. and MILLER, G. (1998). (eds.) *WordNet*. The MIT Press.
- GENESERETH, M. and NILSSON, N. (1987) : *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann, chap. 7 Induction, pages 161-176.
- HARABAGIU, S. and MOLDOVAN, D. (1998). Knowledge processing on an extended wordnet. *WordNet : An Electronic Lexical Database*, MIT Press, pages 381-405.
- LAFOURCADE, M. et JOUBERT, A. (2013) : Bénéfices et limites de l'acquisition lexicale dans l'expérience JeuxDeMots. In *Ressources Lexicales : Contenu, construction, utilisation, évaluation, Linguisticae Investigationes, Supplementa 30*, John Benjamins, pages 187-216.
- LAFOURCADE, M., LE BRUN, N., et JOUBERT, A. (2015) *Jeux et intelligence collective - résolution de problèmes et acquisition de données sur le Web*. Collection Science cognitive et management des connaissances (sous la direction de Joseph Mariani et Patrick Paroubek), ISTE éditions, 2015, 156 p.
- LAFOURCADE, M. (2007). Making people play for lexical acquisition. In *Proc. SNLP 2007, 7th Symposium on Natural Language Processing*. Pattaya, Thaïlande, 13-15 December 2007, 8 p.
- LENAT, D. (1995). Cyc : A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11), pages 33-38.
- LIEBERMAN, H., SMITH, D. A. et TEETERS, A. (2007). Common consensus : a web-based game for collecting commonsense goals. In *Proc. of IUI, Hawaii.*, 12 p.
- MARCHETTI, A., TESCONI, M., RONZANO, F., MOSELLA, M. et MINUTOLI, S. (2007). Semkey : A semantic collaborative tagging system. in *Procs of WWW2007, Banff, Canada*, 9 p.
- MIHALCEA, R. et CHKLOVSKI, T. (2003). Open mindword expert : Creating large annotated data collections with web users help. In *Proceedings of the EACL 2003, Workshop on Linguistically Annotated Corpora (LINC 2003)*, 10 p.
- MILLER, G., BECKWITH, R., FELLBAUM, C., GROSS, D. et MILLER, K. (1990). *Introduction to*

- wordnet : an on-line lexical database. *International Journal of Lexicography*, 3(4), pages 235–244.
- NAVIGLI, R. and PONZETTO, S. (2012). Babelnet : Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, 11-16 July 2010, pages 216–225.
- SAGOT, B. et FIER, D. (2008). Construction d'un wordnet libre du français à partir de ressources multilingues. *TALN 2008*, Avignon, France, 2008, 12 pages.
- von AHN, L. and DABBISH, L. (2008). Designing games with a purpose. *Communications of the ACM*, 51(8), pages 58–67.
- VOSSSEN, P. (1998). *Eurowordnet : a multilingual database with lexical semantic networks*. Kluwer Academic Publishers, Norwell, MA, USA, 200 pages.
- ZARROUK, M., LAFOURCADE, M and JOUBERT, A. (2013) Inference and reconciliation in a lexical-semantic network. In *proc of 14th International Conference on Intelligent Text Processing and Computational Linguistic (CICLING-2013)*, University of the Aegean, Samos, Greece, March 24–30, 2013, 13 p.
- ZARROUK, M. and LAFOURCADE, M. (2014) : Inferring knowledge with word refinements in a crowdsourced lexical semantic network. In *Proc. of the the 25th International Conference on Computational Linguistics (COLING 2014)*, pages 346–366. Dublin, Irlande, 2014, 9 p.
- ZEICHNER, N., BERANT, J. and DAGAN, I. (2012) Crowdsourcing Inference - Rule Evaluation. *ACL 2012*, 5 p.
- ZESCH, T. and GUREVYCH, I. (2009). *Wisdom of crowds versus wisdom of linguists measuring the semantic relatedness of words*. *Natural Language Engineering*, Cambridge University Press, pages 25–59.