

# **Si les souris étaient des reptiles, alors les reptiles pourraient être des mammifères ou Comment détecter les anomalies dans un réseau lexical ?**

Mathieu Lafourcade<sup>1</sup> Alain Joubert<sup>1</sup> Nathalie Le Brun<sup>2</sup>  
(1) LIRMM, 860 rue de St Priest, 34095 Montpellier cedex 5, France  
(2) Imagin@t, 34400 Lunel, France

lafourcade@lirmm.fr, joubert@lirmm.fr, [imaginat@imaginat.name](mailto:imaginat@imaginat.name)

## **RESUME**

---

La correction des erreurs dans une collection de données est un problème délicat. Elle peut être réalisée manuellement par un expert, ou en utilisant des méthodes de crowdsourcing, ou encore automatiquement au moyen d'algorithmes. Nous présentons ici des méthodes automatiques permettant de détecter les erreurs potentielles « secondaires » induites par les mécanismes automatiques d'inférences de relations, lorsqu'ils s'appuient sur des relations erronées « initiales » détectées manuellement. Des résultats encourageants, mesurés sur le réseau JeuxDeMots, nous invitent à envisager également des stratégies qui permettraient de détecter automatiquement les relations erronées « initiales », ce qui pourrait conduire à une détection automatique de la majorité des erreurs présentes dans le réseau.

## **ABSTRACT**

---

**If mice were reptiles, then the reptiles could be mammals, or How to detect errors in a lexical network?**

Correcting errors in a data set is a critical issue. This task can be either hand-made by experts, or by crowdsourcing methods, or automatically done using algorithms. We present here automatic methods for detecting potential "secondary" errors that would result from automatic inference mechanisms when they rely on an "initial" error manually detected. We measured encouraging results on the JeuxDeMots network; they also invite us to consider strategies that would automatically detect erroneous "initial" relations, which could lead to the automatic detection of the majority of errors in the network.

**MOTS-CLES :** Réseau JeuxDeMots, détection des relations erronées, inférence de relations

**KEYWORDS:** JeuxDeMots network, detection of erroneous relations, inference of relations

---

# 1. Introduction

Toute collection de données est nécessairement entachée d'erreurs et, selon le domaine et les applications concernés, leur nombre est plus ou moins tolérable. Les réseaux lexicaux ne font pas exception à la règle : dans le réseau JeuxDeMots (JDM), le taux d'anomalies est relativement faible (largement inférieur à 1%). Minimiser ce taux d'erreur reste toutefois une priorité, et suppose, pour optimiser le taux de correction, des stratégies de détection efficaces.

Les anomalies sont de natures variées. Elles peuvent porter soit sur les termes, comme les fautes d'orthographe (ex : *théâtre/théâtre*), soit sur les relations entre termes (ex : *Milou est\_un être\_humain* ou *Dalida idée\_associée Samson*, par confusion entre *Dalida* et *Dalila*).

Actuellement, la détection des anomalies est essentiellement réalisée manuellement par des joueurs/contributeurs *via* leur activité d'enrichissement du réseau à travers l'interface Diko (le dictionnaire contributif d'associations lexicales du projet JDM). Mais comme elle est fortuite, elle ne peut en aucun cas prétendre à l'exhaustivité, d'où la nécessité de mettre au point une vraie méthode de détection.

Dans un premier temps, nous regarderons d'où peuvent provenir les anomalies, puis nous présenterons une méthode qui permet de signaler comme fausses un certain nombre de relations, à charge pour le validateur humain d'effectuer les corrections correspondantes.

## 2. D'où proviennent les anomalies dans le réseau JDM ?

Le mode de construction et les caractéristiques du réseau JDM tels que décrits dans (Lafourcade *et al.*, 2015) le rendent vulnérable à deux grands types d'erreurs :

- Les anomalies « initiales » introduites par les joueurs et contributeurs, volontairement ou non. Selon notre expérience, il s'agit essentiellement d'erreurs involontaires, car l'intérêt de saisir volontairement des informations erronées est très limité. Effectivement, de par le principe de construction du réseau JDM, une relation n'est créée (ou renforcée si elle existait déjà) que si les deux joueurs l'ont proposée en réponse à une même consigne. Le fait que les parties soient jouées de façon anonyme et asynchrone interdit toute communication entre les deux joueurs d'une même partie. Saisir des incohérences n'a donc qu'une probabilité extrêmement faible d'avoir des conséquences sur les données enregistrées. Cela entraîne une forte diminution du risque, sans toutefois l'annuler totalement.
- Les anomalies « secondaires » induites par les mécanismes automatiques d'inférences (déduction, induction, abduction) à partir des anomalies initiales. Effectivement, le réseau JDM peut être densifié automatiquement par inférences à partir des relations déjà existantes (Lafourcade *et al.*, 2014) ; si certaines de ces relations « initiales » sont erronées, alors les mécanismes d'inférences vont générer des relations potentiellement erronées. Quand les relations inférées automatiquement sont jugées incertaines par le système, (à savoir qu'elles sont détectées par les algorithmes présentés ci-après), alors leur validation est soumise à un processus de votes majoritaires et/ou un avis d'expert, ce qui réduit fortement le risque d'enregistrer des relations erronées.

Prenons un exemple : nous avons la relation *souris is\_a mammifère*. Supposons que soit apparue ensuite la relation erronée « initiale » : *souris is\_a reptile*. Le système dispose d'une série de paires d'hyperonymes incompatibles ; il « sait » donc que *mammifère* et *reptile* sont

incompatibles, tout comme *mammifère* et *poisson* ou *mammifère* et *insecte*, par exemple. Il en déduit alors que le terme *souris* est polysémique, et donc apparaissent les raffinements : *souris* > *mammifère* et *souris* > *reptile*. A partir de ce dernier raffinement, par des mécanismes de déduction/induction, le système peut générer de nouvelles relations qui seront probablement erronées puisque le raffinement *souris* > *reptile* est erroné.

### 3. Comment détecter et corriger les anomalies

Plusieurs auteurs se sont penchés sur la problématique de la détection/correction automatique des erreurs dans le domaine du TALN. (Boudin et Hernandez, 2012) proposent des méthodes de détection/correction automatique des erreurs d'annotation de syntaxe dans le French Treebank, qui s'appuient sur une approche présentée précédemment par (Dikinson et Meurers, 2003). Concernant la détection/correction des erreurs sémantiques, citons les travaux de (Ben Othmane Zribi *et al.*, 2007) pour la langue arabe. (Bouraoui et al, 2009) ont analysé les différents types d'erreurs que l'on rencontre en expression écrite, afin de réaliser une typologie générale des erreurs.

Avant toute chose, signalons que, dans le réseau JDM, corriger une anomalie ne signifie pas effacer la relation concernée, mais plus exactement la *négativer*, c'est-à-dire lui affecter un poids négatif. En effet, il peut être intéressant d'avoir l'information qu'une relation n'est pas vraie, plutôt que de ne pas avoir d'information sur cette relation. Par exemple, la relation *autruche agent voler* pondérée négativement signifie qu'une *autruche* ne peut pas *voler*, alors que l'absence de relation signifierait que l'on ne sait pas si une *autruche* peut *voler* ou non. De plus, en négativant une relation, on s'assure qu'elle ne puisse pas réapparaître, ce qui pourrait être le cas en la supprimant. Notre détecteur automatique d'erreurs va préparer ce processus de correction : en contribuant par un vote négatif sur une relation suspecte, il va la signaler comme telle au validateur humain, qui tranchera en décidant de la négativer (ou pas...).

#### 3.1. Anomalies initiales

Elles semblent difficiles à détecter par des mécanismes endogènes. Si, par exemple deux joueurs sur une même partie ont proposé la relation *souris is\_a reptile*, le système ne n'a pas actuellement les moyens de détecter qu'il s'agit d'une relation erronée. Il peut juste en déduire que le terme *souris* est polysémique. Ces relations sont actuellement signalées par les contributeurs/joueurs puis corrigées manuellement par un expert. Des mécanismes exogènes, basés sur des connaissances extérieures au réseau JDM (Wikipédia, Babelnet ...), pourraient être envisagés, en particulier en ce qui concerne les fautes d'orthographe sur les termes.

#### 3.2. Anomalies secondaires

Quand une anomalie est détectée (par un contributeur), comment retrouver les inférences fausses qu'a pu faire le système à partir de cette erreur ? Il s'avère que ces relations fausses que le système a inférées ont été créées soit par déduction, soit, dans une moindre mesure, par induction (Lafourcade *et al.*, 2014).

##### Déduction

Le mécanisme de la déduction est le suivant : Soit  $R$  une relation sémantique quelconque. Si  $A$  *is\_a*  $B$  et  $B R C$ , alors  $A R C$  possible (sauf exception ou polysémie de  $B$ ). Donc, si  $A$  *is\_a*  $B$  est fausse, il se peut que  $A R C$  le soit aussi. Il faut signaler comme fausses les relations sortantes de  $A$  qui proviennent des propriétés de  $B$ , excepté celles qui proviennent des hyperonymes de  $A$ .

Ex : *souris is\_a reptile* => inférences sur *souris* basées sur les propriétés de *reptile*  
Comment détecter, et donc négativer, ces inférences fausses ? On sait que *souris is\_a mammifère*.  
Dans les relations de *souris*, il faut signaler comme potentiellement fausses celles qui proviennent des propriétés de *reptile* à l'exclusion de celles qui seraient en commun entre *reptile* et *mammifère* :

*reptile r\_agent pondre des œufs*, ce qui n'est pas une relation partagée avec *mammifère* => donc, la relation *souris agent pondre\_des\_œufs* (inférée par déduction) est à signaler comme fausse.  
*a contrario*, *reptile has\_part vertèbres*, et *mammifère has\_part vertèbres* => donc, la relation *souris has\_part vertèbres* est à conserver.

Le problème ne se limite pas aux relations portant sur le seul terme pour lequel l'anomalie a été décelée. Par processus de déduction, le système a pu inférer de nouvelles relations erronées (ex : *souris agent pondre\_des\_œufs*), et à partir de ces relations erronées, inférer de nouvelles relations erronées. La relation erronée partant de *souris*, tout spécifique de *souris* a pu être infecté par application de la déduction. Il sera donc nécessaire de rechercher également des relations « secondaires » potentiellement erronées déduites des spécifiques de *souris*.

Par exemple, comme *souris\_blanche is\_a souris*, la relation *souris\_blanche agent pondre\_des\_œufs* a pu être inférée. Comment détecter cette nouvelle relation erronée ? La relation *souris agent pondre\_des\_œufs* ayant reçu un vote négatif à l'étape précédente le système de détection va en tenir compte et signaler également cette relation comme suspecte en l'affectant d'une contribution négative.

Les anomalies « secondaires » ont pu également se propager dans la chaîne des génériques du terme initial. Si *A is\_a B* est erronée, dans la recherche des relations « secondaires » potentiellement erronées, il est nécessaire qu'un validateur remonte dans la chaîne des génériques de B pour vérifier la validité des différentes relations de généricité entre A et les termes rencontrés. Dès que le validateur rencontrera une relation valide, il ne sera plus nécessaire de continuer à remonter.

- ex : *souris is\_a reptile* est erronée. Donc avons-nous :
- *reptile is\_a sauropside* => *souris is\_a sauropside* ? réponse : relation erronée, on la négative et on continue
- *sauropside is\_a vertébré* => *souris is\_a vertébré* ? réponse : relation valide. On peut donc arrêter le processus de recherche des relations « secondaires » potentiellement erronées trouvées par déduction/induction à *souris is\_a sauropside*.

## Induction

Mécanisme de l'induction : Si *A is\_a B* et *A R C*, alors *B R C* possible (sauf exception ou polysémie de A) Donc, si *A is\_a B* est erronée, il se peut que *B R C* le soit aussi. Il faut donc signaler comme fausses (en les affectant d'un vote négatif) les relations sortantes de B qui proviennent des propriétés de A, à l'exclusion de celles provenant des hyponymes de B.

Ex : *souris is\_a reptile* => inférences sur *reptile* en se basant sur les propriétés de *souris*  
Comment détecter, et donc négativer, ces inférences fausses ?

En procédant par comparaison avec des hyponymes bien renseignés de *reptile* : on sait que *tortue is\_a reptile*. Dans les relations de *reptile*, il faut détecter comme faux ce qui provient des propriétés de *souris*, à l'exclusion de ce qui provient des propriétés de *tortue*.

- *souris has\_part poils*, ce que n'a pas *tortue* ; donc, la relation *reptile has\_part poils* (inférée par induction) est à signaler comme fausse.
- a contrario, *souris has\_part tête*, et *tortue has\_part tête* ; donc, la relation *reptile has\_part tête* est à conserver.

## Algorithmes

Nous synthétisons les considérations ci-dessus sous la forme des deux algorithmes suivants :

```
fonction ListeRelations erroneesParDeduction (Terme A, Z)
// pré-requis : A is_a Z est une relation erronée
// résultat : retourne la liste des relations sortantes de A, présentes dans le réseau JDM,
//             qui sont potentiellement erronées du fait que A is_a Z est erronée
Début
  LR = new ListeRelations()
  LT = hyper (A, Z) // liste des hyperonymes de A non hyperonymes de Z
  Pour chaque B ∈ LT
  Faire   LR = LR ∪ déduitErronées (A, B, Z)
          // liste des relations sortantes de A potentiellement erronées
  FinPour
  Retourne LR
Fin
```

```
fonction ListeRelations erroneesParInduction (Terme A, Z)
// pré-requis : A is_a Z est une relation erronée
// résultat : retourne la liste des relations sortantes de Z, présentes dans le réseau JDM,
//             qui sont potentiellement erronées du fait que A is_a Z est erronée
Début
  LR = new ListeRelations()
  Pour chaque relation de type ARC
  // Parcours sur l'ensemble des relations sortantes de A
  Faire   Si ZRC > 0 // si elle est sortante de Z, il se pourrait qu'elle soit erronée
          Alors   Si erroneeParInduction (A, C, Z) // ZRC potentiellement erronée ?
                  Alors LR = LR ∪ ZRC
                  FinSi
          FinSi
  FinPour
  Retourne LR
Fin
```

Pour ces anomalies secondaires, une question subsiste : les mécanismes endogènes mis en œuvre détectent-ils tout ? Ou plus exactement quelle proportion d'anomalies est corrigée par ces mécanismes ? De plus, ces mécanismes de correction ne risquent-ils pas d'introduire des erreurs ?

### 3.3. Résultats

Pour évaluer les performances de notre système de détection de relations fausses, nous avons (sur une copie locale du réseau lexical JeuxDeMots) artificiellement ajouté des hyperonymes faux à des termes. Nous disposons dans JDM d'une liste de paires d'hyperonymes incompatibles, c'est-à-dire tels qu'un terme ne peut pas avoir les deux termes d'une paire comme génériques, à moins d'être polysémique. Par exemple : *poisson* – *mammifère* ; *insecte* – *reptile* ; *animal* – *plante* ; *avion* – *navire* ; *homme* – *femme* ; *voiture* – *avion* ; *train* – *bateau*, etc. Nous avons sélectionné 250 termes ayant comme hyperonyme un des génériques ci-dessus mais pas le second, que nous avons donc rajouté. Nous avons ensuite lancé sur les 250 termes les mécanismes d'inférences. Nous avons ensuite appliqué nos deux algorithmes de détection de relations fausses afin de trouver les relations à éliminer.

Les mécanismes d'inférences ont produit 4500 nouvelles relations que nous avons évaluées via l'application en ligne Askit (<http://jeuxdemots/askit.php>). Nous avons retenu les 3600 qui ont été évalués au moins deux fois. Le tableau suivant présente les résultats de l'évaluation globalement et pour quelques exemples de couples de génériques incompatibles.

<b>% relations trouvées par les 2 algorithmes</b>	<b>global</b>	<b>poisson / *mammifère</b>	<b>insecte / *reptile</b>	<b>animal / *plante</b>	<b>avion / *navire</b>	<b>homme / *femme</b>
% correctement trouvées	97.6	98.2	95.6	98.4	96.1	99.2
% faux positifs	2.4	1.8	4.4	1.6	3.8	0.8
% faux négatifs	0.52	0.25	0.67	0.35	0.7	0.3

Les termes avec un astérisque (\*) sont les génériques faux introduits afin de produire des inférences erronées. Globalement, nos algorithmes retrouvent 97 % des relations fausses ayant été inférées à partir d'un générique erroné. Ils « loupent » 3% de relations fausses, et trouve 0.5% de faux négatifs (ils supposent fausses des relations qui sont vraies). L'analyse des faux négatifs nous indique qu'il s'agit soit d'exceptions soit de conclusions légitimes vu l'état de complétion du réseau lexical (des relations importantes peuvent manquer).

On remarque des écarts assez importants selon les couples de génériques incompatibles. Manifestement, les différences entre *homme* et *femme* font que les inférences fausses sont plus facilement détectables qu'entre *insecte* et *reptile*. On peut supposer que le réseau est beaucoup plus abondamment et précisément renseigné concernant l'espèce humaine, que des domaines de spécialité comme la zoologie, et que d'une manière générale, les performances diminuent avec le degré de spécialisation du domaine, en raison d'une moindre quantité d'information dans le réseau.

## 4. Conclusion

Bien que le taux d'anomalies dans le réseau JDM soit faible, nous pouvons le diminuer encore par des mécanismes endogènes automatiques de détection des relations erronées. Ces mécanismes, relativement simples, permettent de détecter une proportion importante des relations potentiellement erronées « secondaires » inférées à partir de relations « initiales » fausses. Par ailleurs, comme ces relations signalées comme suspectes sont invalidées dès lors qu'un expert a vérifié qu'elles étaient bien erronées, elles ne peuvent plus donner lieu à de nouvelles inférences fausses et donc servir de terreau à la naissance et à la propagation de nouvelles erreurs, ce qui joue aussi en faveur d'une diminution globale du taux d'erreurs du réseau. Les erreurs "initiales" sont en revanche plus délicates à détecter de manière automatique. Cependant, l'exploitation des listes de génériques incompatibles est une piste intéressante dans la mesure où elle permet de donner l'alerte lorsqu'un terme *a priori* monosémique a deux termes incompatibles pour hyperonymes.

Pour conclure, la synergie entre détection manuelle exercée par les joueurs/contributeurs et méthodes de détection automatiques contribue à maintenir dans le réseau JDM un taux d'erreurs raisonnablement bas.

## Références

BEN OTHMANE ZRIBI C., MEJRI H., BEN AHMED M. (2007). Un analyseur hybride pour la détection et la correction des erreurs cachées sémantiques en langue arabe, *TALN 2007*, Toulouse, 5–8 juin 2007

BOUDIN F., HERNANDEZ N. (2012). Détection et correction automatique d'erreurs d'annotation morpho-syntaxique du French TreeBank. *TALN 2012*, Juin 2012, Grenoble. pp.281-291.

BOURAOUI J.-L., BOISSIERE P., MOJAHID M., VIGOUROUX N., LAGARRIGUE A., VELLA F., NESPOULOUS J.-L. (2009) Problématique d'analyse et de modélisation des erreurs en production écrite. Approche interdisciplinaire. *TALN 2009*, Senlis, 24-26 juin 2009

DICKINSON M., MEURERS W.D. (2003). Detecting errors in part-of-speech annotation. *EACL 2003 (10th Conference of the European Chapter of the Association for Computational Linguistics)*, pp.107–114, Budapest, Hungary

LAFOURCADE M., ZARROUK M., JOUBERT A. (2014). About Inferences in a Crowdsourced Lexical-Semantic Network, *EACL 2014 (14th Conference of the European Chapter of the Association for Computational Linguistics)*, Gothenburg (Sweden), April 2014

LAFOURCADE M., LE BRUN N., JOUBERT A. (2015). *Games with a Purpose (GWAPS)*, ISBN: 978-1-84821-803-1 July 2015, Wiley-ISTE, 158 p.