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

Mathieu Lafourcade[1] Alain Joubert[1] Nathalie Le Brun[2]
(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

Abstract

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. Although even if the rate of errors present in a given lexical network is rather low, it is important to reduce it. 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. Encouraging results 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 a lexical-semantic network.

1 Introduction

Any collection of data contains errors and, depending on the domain and applications concerned, their quantity is more or less tolerable. Although the anomaly rate is relatively low (well below 1%), the JeuxDeMots (JDM) network is no exception (Lafourcade, 2007). Minimizing this error rate remains a priority and requires effective detection strategies to optimize the correction rate.

The anomalies are various. They may relate either to terms, such as spelling mistakes (eg *théâtre / théatre*) or to the relations between terms (eg, *Milou est_un humain*, or *Dalida*, an *idea_associated* to *Samson*, by confusion between *Dalida* and *Delilah*).

Currently, the detection of anomalies is essentially carried out manually by players / contributors via their activity of enriching the network through the Diko interface[1] (the contributory dictionary of lexical associations of the JDM project). But since errors are discovered by chance, this mode of detection cannot claim to be exhaustive, hence the need to develop a true detection method.

First, we investigate the origin of the anomalies and then present a method that detects and reports a number of relationships as false. It is up to the human validator to decide whether to make corrections or not. Although experimented and exemplified on the JDM data, our approach remains fully generic and can be applied on other lexical-semantic network.

2 Where do the Anomalies in a Lexical-Semantic Network Come from?

The construction method and the characteristics of the JDM network (our test case) as described in (Lafourcade *et al.*, 2015) make it vulnerable to two main types of errors:

"Initial" anomalies introduced by the players and the contributors, voluntarily or not. In our experience, these are essentially unintentional errors, as the interest in voluntarily entering erroneous information is very limited. Indeed,

[1] http://www.jeuxdemots.org/diko.php

Proceedings of Recent Advances in Natural Language Processing, pages 424–430,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_056

because of the principle of building the JDM network, a relation is created (or reinforced if it already exists) only if both players have proposed it in response to the same instruction. The games being played anonymously and asynchronously, any communication between the players of the same game is not possible, therefore cheating is made very difficult if not impossible. Entering inconsistencies therefore has only an extremely low probability of having consequences on the recorded data. This leads to a significant reduction in risk, but does not completely cancel it.

"Secondary" anomalies are induced by automatic inference mechanisms (deduction, induction, abduction) from initial anomalies. Indeed, the JDM network can be densified automatically by inferences from existing relations (Lafourcade *et al.*, 2014); if some of these "initial" relations are wrong, then the inference mechanisms will generate potentially erroneous relations. When automatically inferred relations are considered doubtful by the system, their validation is subject to a majority vote process and / or expert opinion, which greatly reduces the risk of recording erroneous relations.

Let's take an example: we have the relation *mouse is_a mammal*. Let us suppose that the erroneous "initial" relation has then appeared: *mouse is_a reptile*. The system "knows" that *mammal* and *reptile* are incompatible, just like *mammal* and *fish* or *mammal* and *insect*, for example. It deduces then that *mouse* is polysemous, and thus appear the refinements: *mouse>mammal* and *mouse>reptile*. From this latter refinement, by deduction / induction mechanisms, the system can generate new relations which will probably be erroneous since the refinement *mouse>reptile* is erroneous.

3 How to Detect and Correct Anomalies

Several authors have studied the problem of automatic detection / correction of errors in the domain of NLP. Boudin and Hernandez (2012) propose methods for automatic detection / correction of syntax annotation errors in the French Treebank, based on an approach previously presented by Dikinson and Meurers (2003). Regarding the detection / correction of semantic errors, we find the work of Ben Othmane Zribi *et al.* (2007) for the Arabic language. Bouraoui *et al.* (2009) analyzed the different types of errors encountered in written expression, in order to realize a general typology of errors.

First of all, let us note that in the JDM network, correcting an anomaly does not mean erasing the relation concerned, but more precisely negating it, that is assigning it a negative weight. A relation with a negative weight is deemed to be false.

Indeed, it may be interesting to have (or keep) the information that a relation is not true, rather than having no information about that relation (by deleting it).

For example, if the relation *ostrich* r_agent *fly* is negatively weighted it means that an ostrich cannot fly, while the absence of relation would mean that one does not know whether an ostrich can fly or not.

Moreover, by negating a relation, one ensures that it cannot reappear, which could be the case by suppressing it. Our automatic error detector will prepare this correction process: by contributing by a negative vote on a suspicious relation, it will signal it as such to the human validator, who will decide and negate it (or not ...).

3.1 Initial Anomalies

Initial errors seem difficult to detect by endogenous mechanisms. For example, if two players on the same game proposed the relation *mouse is_a reptile*, the system is not yet able to detect that it is an erroneous relation. It can only deduce from it that *mouse* is polysemous. These relations are currently reported by contributors / players, then they are manually corrected by an expert (again by negating the relations). Exogenous mechanisms, based on knowledge external to the JDM network (Wikipedia, Babelnet …), could be envisaged, especially with regard to misspelling on terms.

3.2 Secondary Anomalies

When an error is detected (by a contributor), how to find false inferences that the system could make from this error? It turns out that

these false relations that the system has inferred have been created either by deduction, or, to a lesser extent, by induction (Lafourcade *et al*., 2014).

Deduction

The mechanism of deduction is the following (Zarrouk, *et al.*, 2014): let be *R* an arbitrary semantic relation. If *A is_a B* and *B R C*, then *A R C* may be possible (except for exception or polysemy of *B*). Thus, if *A is_a B* is false, it may be that *A R C* is also false. It is necessary to indicate as false the outgoing relations from *A* which come from properties of *B*, except those which come from hypernyms of *A*.

Ex: *mouse is_a reptile* => inferences about *mouse* based on *reptile* properties.
(The statement "*mouse is_a reptile* => inferences" is to be read as "***if*** mouse is a reptile ***then*** some inferences are true")

How to detect, and thus negate, these false inferences? It is known that *mouse is_a mammal*. In the outgoing relations of *mouse*, it is necessary to point out as potentially false those which come from properties of *reptile* with the exclusion of those that would be in common between *reptile* and *mammal*.

For example, *reptile r_agent to_lay_eggs*, is not a shared relation with *mammal* => thus, the relation *mouse agent to_lay_eggs* (inferred by deduction) is to be reported as false.
A contrario, *reptile has_part vertebrae*, and *mammal has_part vertebrae* => thus, the relation *mouse has_part vertebrae* is to be preserved.

The problem is not limited to the relations concerning the only term for which the anomaly was detected. By deduction process, the system was able to infer new erroneous relations (ex: *mouse agent to_lay_eggs*), and from these erroneous relations, infer new erroneous relations.

The erroneous relation being derived from *mouse*, any specific of *mouse* (like *white mouse* or *laboratory mouse*) would be able to be infected by application of the deduction. It will therefore be necessary to also look for potentially erroneous "secondary" relations deduced from some specifics of *mouse*.

For example, as *white_mouse is_a mouse*, the relation *white_mouse agent to_lay_eggs* could be inferred. How to detect this new erroneous relation? The relation *mouse agent to_lay_eggs* having received a negative vote in the preceding step, the detection system will take this into account and also report this relation as suspicious by assigning it a negative contribution.

Secondary errors may also have spread to the generic chain of the initial term. If *A is_a B* is wrong, in order to search for potentially erroneous "secondary" relations, the human validator must go back in the generic chain of B to check the validity of the different generic relations between A and the terms encountered. As soon as the validator encounters a valid relation, it will no longer be necessary to go back up. For example, *mouse is_a reptile* is wrong. So, do we have :

- *reptile is_a sauropside* => *mouse is_a sauropside* ? answer : erroneous relation, it is put negative and the validator goes on
- *sauropside is_a vertebrate* => *mouse is_a vertebrate* ? answer : valid relation. Thus we can stop the process of searching for potentially erroneous "secondary" relations found by deduction/induction with *mouse is_a sauropside*.

Induction

Induction mechanism (for any R relation type): if *A is_a B* and *A R C*, then *B R C* may be possible (except for special cases or polysemy of *A*). Thus, if *A is_a B* is wrong, it is possible that *B R C* is also wrong. The outgoing relations of B which originate from the properties of A, with the exception of those derived from the hyponyms of B, must therefore be indicated as false (by affecting them with a negative vote).

Example: let the wrong relation *mouse is_a reptile* => inferences on *reptile* based on the properties of *mouse*

How to detect these false inferences, so as to negate them?

By comparing with the hyponyms of *reptile* that have a lot of information: we know that *tortoise is_a reptile*. Among reptile relations, those derived from *mouse* properties, excluding those from *turtle* properties should be reported as false.

- *mouse has_part hairs*, which is not the case with the *tortoise* => thus, the relation *reptile has_part hairs* (inferred by induction) has to be reported as false.

- on the opposite, *mouse has_part head*, and *tortoise has_part head* => thus, the relation *reptile has_part head* must be retained.

Algorithms

The following two algorithms (Algorithm 1 and 2) reflect the principles developed above. The first algorithm is related to the deduction mechanisms. The second one is related to induction.

The principle of these algorithms is to virtually simulate what could have been specifically deduced (for algorithm 1) or induced (for algorithm 2) and to eliminate what seems to be incompatible.

Note that a relation that is not in the lexical network has a (virtual) weight equal to 0. The hyper function (resp. hypo) returns the list of hypernyms (resp. hyponyms) of the term given as parameter.

For these secondary anomalies, one question remains: do the implemented endogenous mechanisms used detect everything? Or, more precisely, what proportion of anomalies is detected by these mechanisms? Moreover, are these correction mechanisms not likely to introduce errors, indicating as false some relations that are true?

3.3 Actual and Experimental Results

To evaluate the performance of our false relations detection system, we have (on a local copy of the lexical network JeuxDeMots) artificially added false hypernyms to terms.

In fact, we applied our algorithm on the actual JDM data, in order to correct many errors, which was done successfully. But to perform an evaluation in a controlled environment, we made a copy in which we artificially introduced errors.

On the actual data, we were able to halve the number of errors (from 1% to 0.5%). Most of the remaining errors do not fall into the scope of our proposed method, as they were not detectable through hypernym incompatibilities. For those remaining errors, other approaches should be devised.

We verified manually and in depth that the experiments in controlled environment were not biased by the artificial addition of errors. The actual errors and the errors artificially added are of the same nature. We just added much more numerous and various errors in order to assess our method.

We have in JDM a list of pairs of incompatible hypernyms: this means that a given term cannot have the two terms of a pair from this list as generic, unless it is polysemous. For example: *fish - mammal*; *insect - reptile*; *animal - plant*; *plane - ship*; *man - woman*; *car - plane*; *train - boat*, etc. We selected 250 terms having as hypernym one of the generics above but not the second, which we added. We then launched on these 250 terms the mechanisms of inferences. We then applied on this sample of 250 terms our algorithms of detection of false relations to detect the relations to be eliminated.

The inference mechanisms produced 4,500 new relationships that we evaluated through the Askit [2] online application. We retained the 3,600 new relations that were evaluated at least twice. The following board presents the results of the evaluation, globally and for some examples of couples of incompatible generics.

[2] http://www.jeuxdemots.org/askit.php

% relations found by both algorithms	global	fish / *mammal	insect / *reptile	animal / *plant	plane / *ship	man / *woman
% correctely found	**97.6**	98.2	95.6	98.4	96.1	99.2
% false positives	**2.4**	1.8	4.4	1.6	3.8	0.8
% false negatives	**0.52**	0.25	0.67	0.35	0.7	0.3

Table 1: Evaluation of found relations with both algorithms according to hypernym terms. The first term is the correct one and the second is the one being invalidated. For the first two lines, the sum of each column is 100% as this refers to the amount of false relations. The last line is the percentage of correct relations that are supposed false by our algorithms.

The terms with a star (*) are the false generics introduced in order to produce erroneous inferences. Overall, our algorithms recover 97.6% of the false relations that have been inferred from an erroneous generic. They "miss" 2.4% of false relationships, and find 0.52% false negatives (i.e. they assume as false some relations that are true). The analysis of false negatives indicates that these are either exceptions or relevant conclusions given the state of completion of the lexical network (important relations may be missing).

Significant differences can be observed depending on the pairs of incompatible generics. Obviously, differences between *man* and *woman* make false inferences more easily detectable than between *insect* and *reptile*. It can be assumed that the network is much more extensively and precisely informed about the human species than about areas of specialty such as zoology; we also notice that in general, performance decreases with the degree of specialization of the field, due to the lesser information in the network.

4 Conclusion

Although the rate of anomalies in the JDM network is low, we can reduce it further by automatic endogenous mechanisms for detecting erroneous relationships. These relatively simple mechanisms make possible to detect a significant proportion of the potentially false "secondary" relations inferred from false "initial" relations. Moreover, since these relations, which are reported as suspicious, are invalidated once an expert has verified that they are erroneous, they can no longer give rise to new false inferences and thus serve as a breeding ground for the birth and spread of new errors; this also favours an overall decrease in the network error rate. On the other hand, "initial" errors are more difficult to detect automatically. However, the use of incompatible generic lists is an interesting lead insofar as it allows the alert to be given when a monosemic term has two incompatible terms as hypernyms.

The method we propose can be applied to any lexico-semantic network whose structure is similar to that of JDM. As our approach relies on quite common relation types (hypernym, hyponym, etc.), it is valid for well known lexical resources like WordNet (Miller, 1995), and HowNet (Dong and Dong, 2006) to cite a few. Moreover, our method is independent of the language because it relies only on relations of a semantic nature.

To conclude, the synergy between manual detection by players/contributors and automatic detection methods helps to maintain a reasonably low error rate in the JDM network. Such methods could be applied with great benefit for other resources.

```
function RelationList erroneousByDeduction (Term A, Term Z)
      // required: A is_a Z is an erroneous relation
      // result: list of the outgoing relations from A, being in the JDM network,
      //          which are potentially erroneous because A is_a Z is erroneous
Begin
      LR = new RelationList()
      LT = hyper (A, Z)              // list of hypernyms of A, but not hypernyms of Z
      For each B ∈ LT
      Do       LR = LR ∪ deduceErroneous (A, B, Z)
               // list of relations outgoing from A which are potentially erroneous
      EndFor
      Return LR
End

function RelationList deduceErroneous (Term A, B, Z)
      // required: A is_a B > 0 ; A is_a Z is an erroneous relation ; Z is_a B <= 0
      // result: list of outgoing relations from A, being in the JDM network,
      //          which are potentially erroneous because A is_a Z is erroneous.
      /          B is an hypernym of A that the algorithm uses to detect potentially erroneous relations.
Begin
      L = new RelationList()
      For each relation such as ZRY                  // we check all the outgoing relations from Z
      Do       If  BRY <= 0                          // BRY does not exist or is negative weighted
               Then     If ARY > 0                   // ARY exists (positive weighted)
                        Then L = L ∪ ARY
                        Endif
               EndIf
      EndFor
      Return L
End
```

Algorithm 1: return a list of erroneous relations from a deductive point of view.

```
function RelationList erroneousByInduction (Term A, Term Z)
      // required: A is_a Z is an erroneous relation
      // result: list of the outgoing relations from Z, being in the JDM network,
      //          which are potentially erroneous because A is_a Z is erroneous
Begin
      LR = new RelationList()
      For each relation such as ARC
                                                     // we check all the outgoing relations from A
      Do       If  ZRC > 0                           // if it is outgoing from Z, it may be erroneous
               Then     If erroneousByInduction (A, C, Z)   // is ZRC potentially erroneous?
                        Then LR = LR ∪ ZRC
                        EndIf
               EndIf
      EndFor
      Return LR
End

function boolean isErroneousByInduction (Term A, C, Z)
      // required: ARC > 0 ; A is_a Z is an erroneous relation ; ZRC > 0
      // result: return True if and only if the relation ZRC, being in the JDM network,
      //          is potentially erroneous because A is_a Z is erroneous. C is the target term of an outgoing relation of A
Begin
      LT = hypo (Z,A)                                // list of hyponyms of Z, except A
      Term W = first_term (LT)
      While    W exists and then WRC <= 0            // we check hyponyms of Z
               Do       W = next_term (LT)
      EndWhile
      Return W does not exist
End
```

Algorithm 2: return a list of erroneous relations from an inductive point of view. The *erroneousByInduction* function makes use of the Boolean function *isErroneousByInduction*.

References

BEN OTHMANE ZRIBI C., MEJRI H., AND 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., AND 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., AND 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., AND 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.

DONG Z. AND DONG Q. (2006) *Hownet and the Computation of Meaning*. World Scientific Publishing Co., Inc., River Edge, NJ, USA.

LAFOURCADE M. (2007) *Making people play for Lexical Acquisition.* In Proc. SNLP 2007, 7th Symposium on Natural Language Processing. Pattaya, Thailande, 13-15 December 2007, 8 p.

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

MILLER G A. (1995) Wordnet: A Lexical Database for English. *Communications of the ACM*. Vol. 38, No. 11: 39-41.

ZARROUK M., LAFOURCADE M., AND 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

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), Dublin, Irlande, 9 p.