About Inferences in a Crowdsourced Lexical-Semantic Network

Manel Zarrouk
UM2-LIRMM
161 rue Ada
34095 Montpellier, FRANCE
manel.zarrouk@lirmm.fr

Mathieu Lafourcade
UM2-LIRMM
161 rue Ada
34095 Montpellier, FRANCE
mathieu.lafourcade@lirmm.fr

Alain Joubert
UM2-LIRMM
161 rue Ada
34095 Montpellier, FRANCE
alain.joubert@lirmm.fr

Abstract

Automatically inferring new relations from already existing ones is a way to improve the quality of a lexical network by relation densification and error detection. In this paper, we devise such an approach for the JeuxDeMots lexical network, which is a freely avalaible lexical network for French. We first present deduction (generic to specific) and induction (specific to generic) which are two inference schemes ontologically founded. We then propose abduction as a third form of inference scheme, which exploits examples similar to a target term.

1 Introduction

Building resources for Computational Linguistics (CL) is of crucial interest. Most of existing lexical-semantic networks have been built by hand (like for instance WordNet (Miller et al., 1990)) and, despite that tools are generally designed for consistency checking, the task remains time consuming and costly. Fully automated approaches are generally limited to term co-occurrences as extracting precise semantic relations between terms from corpora remains really difficult. Meanwhile, crowdsourcing approaches are flowering in CL especially with the advent of Amazon Mechanical Turk or in a broader scope Wikipedia and Wiktionary, to cite the most well-known examples. WordNet is such a lexical network, constructed by hand at great cost, based on synsets which can be roughly considered as concepts (Fellbaum, 1988). EuroWordnet (Vossen., 1998) a multilingual version of WordNet and WOLF (Sagot., 2008) a French version of WordNet, were built by automated crossing of WordNet and other lexical resources along with some manual checking. Navigli (2010) constructed automatically BabelNet a large multilingual lexical network from term co-occurrences in Wikipedia.

A lexical-semantic network can contain lemmas, word forms and multi-word expressions as entry points (nodes) along with word meanings and concepts. The idea itself of *word senses* in the lexicographic tradition may be debatable in the context of resources for semantic analysis, and we generally prefer to consider *word usages*. A given polysemous word, as identified by locutors, has several usages that might differ substantially from word senses as classically defined. A given usage can also in turn have several deeper refinements and the whole set of usages can take the form of a decision tree. For example, *frigate* can be a bird or a ship. A *frigate>boat* can be distinguished as a modern ship with missiles and radar or an ancient vessel with sails. In the context of a collaborative construction, such a lexical resource should be considered as being constantly evolving and a general rule of thumb is to have no definite certitude about the state of an entry. For a polysemic term, some refinements might be just missing at a given time notwithstanding evolution of language which might be very fast, especially in technical domains. There is no way (unless by inspection) to know if a given entry refinements are fully completed, and even if this question is really relevant.

The building of a collaborative lexical network (or, in all generality, any similar resource) can be devised according to two broad strategies. First, it can be designed as a contributive system

Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 174–182,
Gothenburg, Sweden, April 26-30 2014.

like Wikipedia where people willingly add and complete entries (like for Wiktionary). Second, contributions can be made indirectly thanks to games (better known as GWAP (vonAhn, 2008)) and in this case players do not need to be aware that while playing they are helping building a lexical resource. In any case, the built lexical network is not free of errors which are corrected along their discovery. Thus, a large number of obvious relations are not contained in the lexical network but are indeed necessary for a high quality resources usable in various NLP applications and notably semantic analysis. For example, contributors seldom indicate that a particular bird type can fly, as it is considered as an obvious generality. Only notable facts which are not easily deductible are naturally contributed. Well known exceptions are also generally contributed and take the form of a negative weight and annotated as such (for example, *fly* $\xrightarrow{agent:-100}$ *ostrich* [exception: bird]).

In order to consolidate the lexical network, we adopt a strategy based on a simple inference mechanism to propose new relations from those already existing. The approach is strictly endogenous (i.e. self-contained) as it doesn't rely on any other external resources. Inferred relations are submitted either to contributors for voting or to experts for direct validation/invalidation. A large percentage of the inferred relations has been found to be correct however, a non-negligible part of them are found to be wrong and understanding why is both interesting and useful. The explanation process can be viewed as a *reconciliation* between the inference engine and contributors who are guided through a dialog to explain why they found the considered relation incorrect. The possible causes for a wrong inferred relation may come from three possible origins: false premises that were used by the inference engine, exception or confusion due to some polysemy.

In (Sajous et al., 2013) an endogenous enrichment of Wiktionary is done thanks to a crowdsourcing tool. A quite similar approach of using crowdsourcing has been considered by (Zeichner, 2012) for evaluating inference rules that are discovered from texts. In (Krachina, 2006), some specific inference methods are conducted on text with the help of an ontology. Similarly, (Besnard, 2008) capture explanation with ontology-based inference. OntoLearn (Velardi, 2006) is a system that automatically build ontologies of specific domains from texts and also makes use of inferences. There have been also researchs on taxonomy induction based on WordNet (Snow, 2006). Although extensive work on inference from texts or handcrafted resources has been done, almost none endogenously on lexical network built by the crowds. Most probably the main reason of that situation is the lack of such specific resources.

In this article, we first present the principles behind the lexical network construction with crowdsourcing and *games with a purpose* (also know as human-based computation games) and illustrated them with the JeuxDeMots (JDM) project. Then, we present the outline of an *elicitation engine* based on an *inference engine* using deduction, induction and especially abduction schemes. An experimentation is then presented.

2 Crowdsourced Lexical Networks

For validating our approach, we used the JDM lexical network, which is constructed thanks to a set of associatory games (Lafourcade, 2007) and has been made freely available by its authors. There is an increasing trend of using online GWAPs (game with a purpose (Thaler et al., 2011)) method for feeding such resources. Beside manual or automated strategies, contributive approaches are flowering and becoming more and more popular as they are both cheap to set up and efficient in quality.

The network is composed of terms (as vertices) and typed relations (as links between vertices) with weight. It contains terms and possible refinements. There are more than 50 types of relations, that range from ontological (hypernym, hyponym), to lexical-semantic (synonym, antonym) and to semantic role (agent, patient, instrument). The weight of a relation is interpreted as a strength, but not directly as a probability of being valid. The JDM network is not an ontology with some clean hierarchy of concepts or terms. A given term can have a substantial set of hypernyms that covers a large part of the ontological chain to upper concepts. For example, hypernym(cat) = {feline, mammal, living being, pet, vertebrate, ...}. Heavier weights associated to relations are those felt by users as being the most relevant. The

1st January 2014, there are more than 6 700 000 relations and roughly 310 000 lexical items in the JDM lexical network (according to the figures given by the game site: http://jeuxdemots.org).

To our knowledge, there is no other existing freely available crowdsourced lexical-network, especially with weighted relations, thus enabling strongly heuristic methods.

3 Inferring with Deduction & Induction

Adding new relations to the JDM lexical network may rely on two components: (a) an inference engine and (b) a reconciliator. The inference engine proposes relations as a contributor to be validated by other human contributors or experts. In case of invalidation of an inferred relation, the reconciliator is invoked to try to assess why the inferred relation was found wrong. Elicitation here should be understood as the process to transform some implicit knowledge of the user into explicit relations in the lexical network. The core ideas about inferences in our engine are the following:

- inferring is to derive new premises (as relations between terms) from previously known premises, which are existing relations;
- candidate inferences may be logically blocked on the basis of the presence or the absence of some other relations;
- candidate inferences can be filtered out on the basis of a strength evaluation.

3.1 Deduction Scheme

Inferring by deduction is a top-down scheme based on the transitivity of the relation *is-a* (hypernym). If a term A is a kind of B and B holds some relation R with C, then we can expect that A holds the same relation type with C. The scheme can be formally written as follows: $\exists\ A \xrightarrow{is-a} B \ \wedge\ \exists\ B \xrightarrow{R} C \ \Rightarrow\ A \xrightarrow{R} C.$

For example, *shark* $\xrightarrow{is-a}$ *fish* and *fish* $\xrightarrow{has-part}$ *fin*, thus we can expect that *shark* $\xrightarrow{has-part}$ *fin*. The inference engine is applied on terms having at least one hypernym (the scheme could not be applied otherwise). Of course, this scheme is far too naive, especially considering the resource we are dealing with and may produce wrong relations (noise). In effect, the central term B is possibly polysemous and ways to avoid probably wrong inferences can be done through a *logical blocking*: if there are two distinct meanings for B that hold respectively the first and the second relation, then most probably the inferred relation R(3) is wrong (see figure 1) and hence should be blocked. Moreover, if one of the premises is tagged by contributors as *true but irrelevant*, then the inference is blocked.

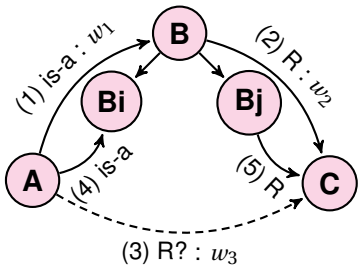

Figure 1: Triangular inference scheme where the logical blocking based on the polysemy of the central term B which has two distinct meanings B_i and B_j is applied. The two arrows without label are those of word meanings.

It is possible to evaluate a confidence level (on an open scale) for each produced inference, in a way that dubious inferences can be eliminated out through *statistical filtering*. The weight w of an inferred relation is the geometric mean of the weight of the premises (relations (1) and (2) in Figure 1). If the second premise has a negative value, the weight is not a number and the proposal is discarded. As the geometric mean is less tolerant to small values than the arithmetic mean, inferences which are not based on two rather strong relations (premises) are unlikely to pass.

$$w(A \xrightarrow{R} C) = (\ w(A \xrightarrow{is-a} B) \times w(B \xrightarrow{R} C)\ )^{1/2}$$
$$\Rightarrow \quad w3 = (w1 \times w2)^{1/2}$$

Inducing a transitive closure over a knowledge base is not new, but doing so considering word meanings over a crowdsourced lexical network is an original approach.

3.2 Induction Scheme

As for the deductive inference, induction exploits the transitivity of the relation *is-a*. If a term A is a kind of B and A holds a relation R with C, then we might expect that B could hold the same type of relation with C. More formally we can write: $\exists\, A \xrightarrow{is-a} B \ \wedge\ \exists\, A \xrightarrow{R} C \ \Rightarrow\ B \xrightarrow{R} C.$ For example, *shark* $\xrightarrow{is-a}$ *fish* and *shark* $\xrightarrow{has-part}$ *jaw*, thus we might expect that *fish* $\xrightarrow{has-part}$ *jaw*.

This scheme is a generalization inference. The principle is similar to the one applied to the de-

duction scheme and similarly some logical and statistical filtering may be undertaken.

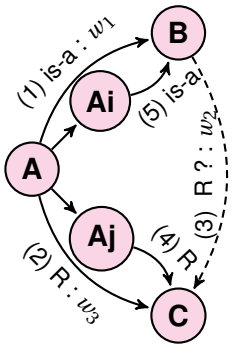

Figure 2: (1) and (2) are the premises, and (3) is the induction proposed for validation. Term A may be polysemous with meanings holding premises, thus inducing a probably wrong relation.

The central term here A, is possibly polysemous (as shown in Figure 2). In that case, we have the same polysemy issues than with the deduction, and the inference may be blocked. The estimated weight for the induced relation is:

$$w(B \xrightarrow{R} C) = (w(A \xrightarrow{R} C))^2 \ / \ w(A \xrightarrow{is-a} B)$$
$$\Rightarrow \quad w2 = (w_3)^2 / w_1$$

3.3 Performing Reconciliation

Inferred relations are presented to the validator to decide of their status. In case of invalidation, a reconciliation procedure is launched in order to diagnose the reasons: error in one of the premises (previously existing relations are false), exception or confusion due to polysemy (the inference has been made on a polysemous central term). A dialog is initiated with the user (Cohen's kappa of 0.79). To know in which order to proceed, the reconciliator checks if the weights of the premises are rather strong or weak.

Errors in the premises. We suppose that relation (1) (in Figure 1 and 2) has a relatively low weight. The reconciliation process asks the validator if the relation (1) is true. It sets a negative weight to this relation if not so that the engine blocks further inferences. Else, if relation (1) is true, we ask about relation (2) and proceed as above if the answer is negative. Otherwise, we check the other cases (exception, polysemy).

Errors due to Exceptions. For the *deduction*, in case we have two trusted relations, the reconciliation process asks the validators if the inferred relation is a kind of exception relatively to the term B. If it is the case, the relation is stored in the lexical network with a negative weight and annotated as *exception*. Relations that are exceptions do not participate further as premises for deducing. For the *induction*, in case we have two trusted relations, the reconciliator asks the validators if the relation ($A \xrightarrow{R} C$) (which served as premise) is an exception relatively to the term B. If it is the case, in addition to storing the false inferred relation ($B \xrightarrow{R} C$) in the lexical network with a negative weight, the relation ($A \xrightarrow{R} C$) is annotated as exception. In the induction case, the exception is a true premise which leads to a false induced relation. In both cases of induction and deduction, the *exception* tag concerns always the relation ($A \xrightarrow{R} C$). Once this relation is annotated as an exception, it will not participate as a premise in inferring generalized relations (bottom-up model) but can still be used in inducing specified relations (top-down model).

Errors due to Polysemy. If the central term (B for deduction and A for induction) presenting a polysemy is mentioned as polysemous in the network, the refinement terms $term_1$, $term_2$, $\ldots term_n$ are presented to the validator so she/he can choose the appropriate one. The validator can propose new terms as refinements if she/he is not satisfied with the listed ones (inducing the creation of new appropriate refinements). If there is no meta information indicating that the term is polysemous, we ask first the validator if it is indeed the case. After this procedure, new relations will be included in the network with positive values and the inference engine will use them later on as premises.

4 Abductive Inference

The last inferring scheme is built upon abduction and can be viewed as an example based strategy. Hence abduction relies on similarity between terms, which may be formalized in our context as sharing some outgoing relations between terms. The abductive inferring layout supposes that relations held by a term can be proposed to similar terms. Here, abduction first selects a set of similar terms to the target term A which are considered as proper examples. The outgoing relations from the examples which are not common with those of A are proposed as potential relations for A and then presented for validation/invalidation to users. Unlike induction and deduction, abduction can be applied on

terms with missing or irrelevant ontological relations, and can generate ontological relations to be used afterward by the inference loop.

4.1 Abduction Scheme

We note an *outgoing relation* as a 3-uple of a type t, a weight w and a target node n:$R_i = \langle\ t_i\ ,\ w_i\ ,\ n_i\ \rangle$. For example, consider the term A having n outgoing relations. Amongst these relations, we have for example:

• $beak \overset{has-part}{\longleftarrow} A$ & • $nest \overset{location}{\longleftarrow} A$.

We found 3 examples sharing those two relations with the term A:

• $beak \overset{has-part}{\longleftarrow} \{ex_1, ex_2, ex_3\}$

• $nest \overset{location}{\longleftarrow} \{ex_1, ex_2, ex_3\}$

We consider these terms as a set of examples to follow and similar to A. These examples have also other outgoing relations which are proposed as potential relations for A. For example :

• $\{ex_1, ex_2\} \overset{agent-1}{\longrightarrow} fly$ • $\{ex_2\} \overset{carac}{\longrightarrow} colorful$

• $\{ex_1, ex_2, ex_3\} \overset{has-part}{\longrightarrow} feather$

• $\{ex_3\} \overset{agent-1}{\longrightarrow} sing$

We infer that A can hold these relations and we propose them for validation.

• $A \overset{agent-1}{\longrightarrow} fly$? • $A \overset{has-part}{\longrightarrow} feather$?

• $A \overset{carac}{\longrightarrow} colorful$? • $A \overset{agent-1}{\longrightarrow} sing$?

4.2 Abduction Filtering

Applying the abduction procedure crudely on the terms generates a lot of waste as a considerable amount of erroneous inferred relations. Hence, we elaborated a filtering strategy to avoid having a lot of dubious proposed candidates. For this purpose, we define two different threshold pairs. The first threshold pair (δ_1, ω_1) is used to select proper examples x_1,x_2...x_n and is defined as follows:

$$\delta_1 = \max(3, \text{nbogr}(A) \times 0.1) \tag{1}$$

where *nbogr(A)* is the number of outgoing relations from the term A.

$$\omega_1 = \max(25, \text{mwogr}(A) \times 0.5) \tag{2}$$

where *mwogr(A)* is the mean of weights of outgoing relations from A. The second threshold pair (δ_2, ω_2) is used to select proper candidate relations from outgoing relations of the examples R'_1,R'_2...R'_q.

$$\delta_2 = \max(3, \overline{\{x_i\}} \times 0.1) \tag{3}$$

where $\overline{\{x_i\}}$ is the cardinal of the set $\{x_i\}$.

$$\omega_2 = \max(25, \text{mwogr}(\{x_i\}) \times 0.5) \tag{4}$$

where $mwogr(\{x_i\})$ is the mean of weights of outgoing relations from the set of examples x_i.

If a term A is sharing at least δ_1 relations, having a weight over ω_1, of the total of the relations R_1, R_2, ... R_p toward terms T_1, T_2, ... T_p with a group of examples x_1, x_2, ... x_n, we admit that this term has a degree of similarity strong enough with these examples. After building up a set of examples on which we can apply our abduction engine we proceed with the second part of the strategy. If we have at least δ_2 examples x_i holding a specific relation R'_k weighting over ω_2 with a term B_k, more formally $R'_k = \langle\ t\ ,\ w \geq \omega_2, B_k\ \rangle$, we can suppose that the term A may hold this same relation R'_k with the same target term B_k (figure 3).

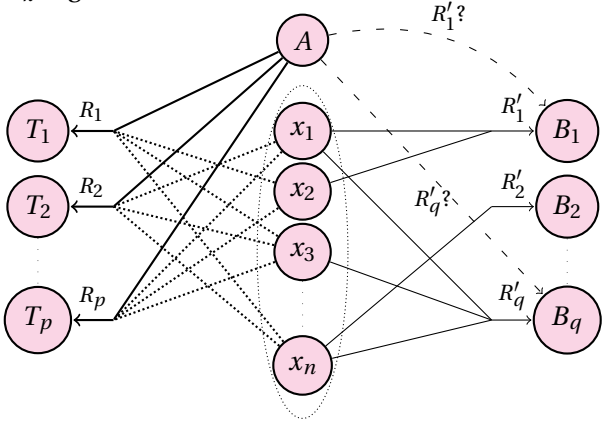

Figure 3: Abduction scheme with examples x_i sharing relations with A and proposing new abducted relations.

On figure 3, we simplified thresholds to 2 for illustrative purpose. So, to be selected, the examples x_1 ,x_2, x_3, ... x_n must have at least 2 common relations with A. A relation $R'_{1\to q}$ must be hold by at least 2 examples to be proposed as a potential relation for A. More clearly:

► $x_1 \overset{R'_1}{\longrightarrow} B_1$ and $x_2 \overset{R'_1}{\longrightarrow} B_1 \Rightarrow R'_1 : 2$

$\Longrightarrow$ propose $A \overset{R'_1?}{\longrightarrow} B_1$

► $x_n \overset{R'_2}{\longrightarrow} B_2 \Rightarrow R'_2 : 1$

$\Longrightarrow$ do not propose this relation.

► $x_1 \overset{R'_q}{\longrightarrow} B_q$, $x_3 \overset{R'_q}{\longrightarrow} B_q$ and $x_n \overset{R'_q}{\longrightarrow} B_q$

$\Rightarrow R'_q : 3$

$\Longrightarrow$ propose $A \overset{R'_q?}{\longrightarrow} B_q$

For statistical filtering, we can act on the

threshold (δ_2, ω_2) as the minimum number of examples x_i being R' related with a target term B_k. It is also possible to evaluate the weight of the abducted relation as following:

$$w(A \xrightarrow{R'_k} B_k) = \frac{1}{\mathrm{nb}_{R'_{cd}}} \sum_{i=1,j=1,k=1}^{n,p,q} \sqrt[3]{w_1 w_2 w_3} \quad (5)$$

where $\mathrm{nb}_{R'_{cd}}$ is the number of the relations R' candidate to be proposed and w_1=$A \xrightarrow{R_j} T_j$ & w_2=$x_i \xrightarrow{R_j} T_j$ & w_3=$x_i \xrightarrow{R'_k} B_k$.

This filtering parameters are adjustable according to the user's requirements, so it can fulfil various expectations. Constant values in threshold formulas have been determined empirically.

5 Experimentation

We made an experiment with a unique run of the deduction, induction and abduction engines over the lexical network. Contributors have either accepted or rejected a subset of those candidates during the normal course of their activity. This experiment is for an evaluation purpose only, as actually the system is running iteratively along with contributors and games. The experiment has been done with the parameters given previously, which are determined empirically as those maximizing recall and precision (over a very small subset of the JDM lexical network, around 1‰).

5.1 Appliying Deductions and Inductions

We applied the inference engine on around 25 000 randomly selected terms having at least one hypernym or one hyponym and thus produced by deduction more than 1 500 000 inferences and produced by induction over 360 000 relation candidates. The threshold for filtering was set to a weight of 25. This value is relevant as when a human contributor proposed relation is validated by experts, it is introduced with a default weight of 25.

The transitive *is-a* (Table1) is not very productive which might seems surprising at first glance. In fact, the *is-a* relation is already quite populated in the network, and as such, fewer new relations can be inferred. The figures are inverted for some other relations that are not so well populated in the lexical network but still are potentially valid. The *has-parts* relation and the agent semantic role (the *agent-1* relation) are by far the most productive types.

Relation type	**Proposed %**
is-a (x is a type of y)	6.1
has-parts (x is composed of y)	25.1
holonym (y specific of x)	7.2
typical place (of x)	7.2
charac (x as characteristic y)	13.7
agent-1 (x can do y)	13.3
instr-1 (x instrument of y)	1.7
patient-1 (x can be y)	1
place-1 (x located in the place y)	9.8
place > action (y can be done in place x)	3.4
object > mater (x is made of y)	0.3

Table 1: Global percentages of relations proposed per type for deduction and induction.

Deduction	**% valid**		**% error**		
Relation type	**rlvt**	**¬ rlvnt**	**prem**	**excep**	**pol**
is-a	76%	13%	2%	0%	9%
has-parts	65%	8%	4%	13%	10%
holonym	57%	16%	2%	20%	5%
typical place	78%	12%	1%	4%	5%
charac	82%	4%	2%	8%	4%
agent-1	81%	11%	1%	4%	3%
instr-1	62%	21%	1%	10%	6%
patient-1	47%	32%	3%	7%	11%
place-1	72%	12%	2%	10%	6%
place > action	67%	25%	1%	4%	3%
object > mater	60%	3%	7%	18%	12%

Table 2: Number of propositions produced by deduction and ratio of relations found as true or false.

In tables 2 and 3 are presented some evaluations of the status of the inferences proposed by the inference engine through deduction and induction respectively. Inferences are valid for an overall of 80-90% with around 10% valid but not relevant (like for instance *dog* $\xrightarrow{has-parts}$ *proton*). We observe that error number in premises is quite low, and nevertheless errors can be easily corrected. Of course, not all possible errors are detected through this process. More interestingly, the reconciliation allows in 5% of the cases to identify polysemous terms and refinements. Globally false negatives (inferences voted false while being true) and false positives (inferences voted true while being false) are evaluated to less than 0.5%.

For the induction process, the relation *is-a* is not obvious (a lexical network is not reductible to an ontology and multiple inheritance is possible). Result seems about 5% better than for the deduction process: inferences are valid for an overall of 80-95%. The error number is very low. The main difference with the deduction process is on errors due to polysemy which is lower with the induction process.

To try to assess a baseline for those results, we compute the full closure of the lexical network, i.e. we produce iteratively all possible candidate relations until no more could be found, each candidate being considered as correct and participating to the process. We got more than 6 000 000 relations out of which 45% were wrong (evaluation on around 1 000 candidates randomly chosen).

5.2 Unleashing the Abductive Engine

We applied systematically the abduction engine on the lexical items contained in the network, and produce 629 987 abducted relations out of which 137 416 were not already existing in the network. Those 137 416 are candidate relations concerning 10 889 distinct lexical entries, hence producing a mean of around 12 new relations per entry. The distribution of the proposed relations follows a power law, which is not totally surprising as the relation distribution in the lexical network is by itself governed by such a distribution. Those figures indicate that abduction seems to be still quite productive in terms of raw candidates, even not relying on ontological existing relations.

The table 4 presents the number of relations proposed by the inference engine through abduction. The different relation types are variously productive, and this is mainly due to the number of existing relations and the distribution of their type. The most productive relation is *has-part* and the least one is *holo* (holonym/whole). Correct relations represent around 80% of the relations that have been evaluated (around 5.6% of the total number of produced relations).

One suprising fact, is that the 80% seem to be quite constant notwithstanding the relation type, the lowest value being 77% (for *instr-1* which is the relation specifying what can be done with *x* as an instrument) and the highest being 85% (for *action-place* which is the relation associating for an action the typical locations where it can occur). The abduction process is not ontologically based, and hence does not rely on the generic (is-a) or specific (hyponym) relations, but on the contrary on any set of examples that seems to be alike the target term. The apparent stability of 80% correct abducted relations may be a positive consequence of relying on a set of examples, with a potentially irreductible of 20% wrong abducted relations.

Figure 4 presents two types of data: (1) the percentage of correct abducted relations according to the number of examples required to produce the inference, and (2) the proportion between the produced relations and the total of 107 416 relations according to the minimal number of examples allowed. What can clearly be seen is that when the number of required examples is increased, the ratio of correct abductions increases accordingly, but the number of proposed relations dramaticaly falls. The number of abductions is an inverse power law of the number of examples required.

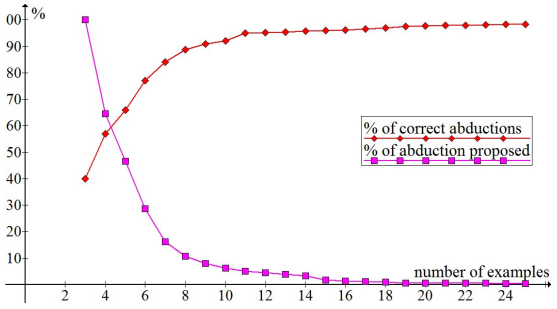

Figure 4: Production of abducted relations and percentage of correctness according to examples number.

At 3 examples, only 40% of the proposed relations are correct, and with a minimum of 6 examples, more than 3/4 of the proposals are deemed correct. The balanced F-score is optimal at the intersection of both curves, that is to say for at least 4 examples.

In figure 5, is showed the mean number of new relations during an iteration of the inference engine on abduction. Between two runs, users and validators are invited to accept or reject abducted relations. This process is done at their discretion and users may leave some propostions unvoted. Experiments showed that users are willing to validate strongly true relations and invalidate clearly false relations. Relations whose status may be difficult are more often left aside than other easiest proposals. The third run is the most productive with a mean of almost 20 new abducted relations. After 3 runs, the abductive process begins to be less productive by attrition of new possible candidates. Notice that the abduction process may, on subsequent runs, remove some previsouly done proposals and as such is not monotonous.

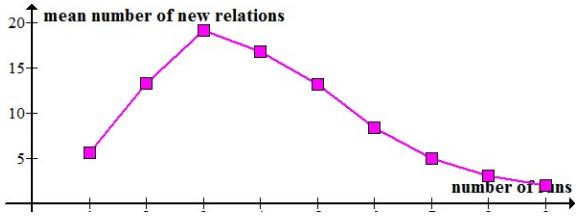

Figure 5: Mean number of new relations relatively to runs in iterated abduction.

5.3 Figures on Reconciliation

Reconciliation in abduction is simpler than in deduction or induction, as the potential adverse effect of polysemy is counterbalanced by the statistical approach implemented by the large number of examples (when available). The reconciliation in the case of abduction is to determine if the wrong proposal has been produced logically considering the support examples. In 97% of the cases, the wrong abducted relation has been qualified as *wrong but logical* by voters or validators. For examples: • *Boeing 747* $\xrightarrow{has-part}$ *propeller** • *whale* $\xrightarrow{place}$ *lake** • *pelican* $\xrightarrow{agent-1}$ *sing **. All those wrong abducted relations given as examples above might have been correct. Considering the examples exploited to produce the candidates, in those cases there is no possible way to guess those relations are wrong. This is even reinforced by the fact that abduction does not rely on ontological relations, which in some cases could have avoided wrong abduction. However, abduction compared to induction and deduction, can be used on terms that do not hold ontological relations, either they are missing or they are not relevant (for verbs, instances...).

6 Conclusion

We presented some issues in inferring new relations from existing ones to consolidate a lexical-semantic network built with games and user contributions. New inferred relations are stored to avoid having to infer them again and again dynamically. To be able to enhance the network quality and coverage, we proposed an elicitation engine based on inferences (induction, deduction and abduction) and reconciliation. If an inferred relation is proven wrong, a reconciliation process is conducted in order to identify the underlying cause and solve the problem. The abduction scheme does not rely on the ontological relation (is-a) but merely on examples that are similarly close to the target term. Experiments showed that abduction is quite productive (compared to deduction and induction), and is stable in correctness. User evaluation showed that wrong abducted relations (around 20% of all abducted relations) are still logically sound and could not have been dismissed *a priori*. Abduction can conclusively be considered as a usefull and efficient tool for relation inference. The main difficulty relies in setting the various parameter in order to achieve a fragile tradeoff between an overrestrictive filter (many false negatives, resulting in information losses) and the opposite (many false postive, more human effort).

The elicitation engine we presented through schemas based on deduction, induction and abduction is an efficient error detector, a polysemy identifier but also a classifier by abduction. The actions taken during the reconciliation forbid an inference proven wrong or exceptional to be inferred again. Each inference scheme is supported by the two others, and if a given inference has been produced by more than one of these three schemas, it is almost surely correct.

Induction	**% valid**		**% error**		
Relation types	**rlvt**	**¬rlvnt**	**prem**	**excep**	**pol**
is-a	-	-	-	-	-
has-parts	78%	10%	3%	2%	7%
holonyme	68%	17%	2%	8%	5%
typical place	81%	13%	1%	2%	3%
charac	87%	6%	2%	2%	3%
agent-1	84%	12%	1%	2%	1%
instr-1	68%	24%	1%	4%	3%
patient-1	57%	36%	3%	2%	2%
place-1	75%	16%	2%	5%	2%
place > action	67%	28%	1%	3%	1%
object > mater	75%	10%	7%	5%	3%

Table 3: Number of propositions produced by induction and ratio of relations found as true or false.

Abduction	**#prop**	**#eval (%)**	**True (%)**	**False (%)**
is-a	7141	421 (5.9)	343 (81.5)	78 (18.5)
has-parts	26517	720 (2.7)	578 (80.3)	142 (19.7)
holo	1592	153 (9.6)	124 (81)	29 (18.9)
agent	7739	298 (3.9)	236 (79.2)	62 (20.8)
place	17148	304 (1.8)	253 (83.2)	51 (16.8)
instr	10790	431 (4)	356 (82.6)	75 (17.4)
charac	7443	319 (4.3)	251 (78.7)	68 (21.3)
agent-1	18147	955 (5.3)	780 (81.7)	175 (18.3)
instr-1	11867	886 (7.5)	682 (77)	204 (23)
place-1	14787	1106 (7.5)	896 (81)	210 (19)
place>act	8268	270 (3.3)	214 (79.3)	56 (20.7)
act>place	5976	170 (2.8)	145 (85.3)	25 (14.7)
Total	137416	6033 (4.3)	4858 (81)	1175 (19)

Table 4: Number of propositions produced by abduction and ratio of relations found as true or false.

References

von Ahn, L. and Dabbish, L. 2008. *Designing games with a purpose.* in Communications of the ACM, number 8, volume 51. p 58-67.

Besnard, P. Cordier, M.-O., and Moinard, Y. 2008. *Ontology-based inference for causal explanation.* Integrated Computer-Aided Engineering , IOS Press, Amsterdam, Vol. 15 , No. 4, 351-367, 2008.

Fellbaum, C. and Miller, G. 1988. *(eds) WordNet.* The MIT Press.

Krachina, O., Raskin, V. 2006. *Ontology-Based Inference Methods.* CERIAS TR 2006-76, 6 p.

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

Lafourcade, M., Joubert, A. 2012. *Long Tail in Weighted Lexical Networks.* In proc of Cognitive Aspects of the Lexicon (CogAlex-III), COLING, Mumbai, India, December 2012.

Lieberman, H, Smith, D. A and Teeters, A 2007. *Common consensus: a web-based game for collecting commonsense goals.* In Proc. of IUI, Hawaii,2007.12 p .

Marchetti, A and Tesconi, M and Ronzano, F and Mosella, M and Minutoli, S. 2007. *SemKey: A Semantic Collaborative Tagging System.* in Procs of WWW2007, Banff, Canada. 9 p.

Mihalcea, R and 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). 10 p.

Miller, G.A. and Beckwith, R. and Fellbaum, C. and Gross, D. and Miller, K.J. 1990. *Introduction to WordNet: an on-line lexical database.* International Journal of Lexicography. Volume 3, p 235-244.

Navigli, R and Ponzetto, S. 2010. *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.p 216-225.

Sagot, B. and Fier, D. 2010. *Construction d'un wordnet libre du français à partir de ressources multilingues.* in Proceedings of TALN 2008, Avignon, France, 2008.12 p.

Sajous, F., Navarro, E., Gaume, B,. Prévot, L. and Chudy, Y. 2013. *Semi-Automatic Enrichment of Crowdsourced Synonymy Networks: The WISIGOTH system applied to Wiktionary.* Language Resources & Evaluation, 47(1), pp. 63-96.

Siorpaes, K. and Hepp, M. 2008. *Games with a Purpose for the Semantic Web.* in IEEE Intelligent Systems, number 3, volume 23.p 50-60.

Snow, R. Jurafsky, D., Y. Ng., A. 2006. *Semantic taxonomy induction from heterogenous evidence.* in Proceedings of COLING/ACL 2006, 8 p.

Thaler, S and Siorpaes, K and Simperl, E. and Hofer, C. 2011. *A Survey on Games for Knowledge Acquisition.* STI Technical Report, May 2011.19 p.

Velardi, P. Navigli, R. Cucchiarelli, A. Neri, F. 2006. *Evaluation of OntoLearn, a methodology for Automatic Learning of Ontologies.* in Ontology Learning and Population, Paul Buitelaar Philipp Cimmiano and Bernardo Magnini Editors, IOS press 2006).

Vossen, P. 2011. *EuroWordNet: a multilingual database with lexical semantic networks.* Kluwer Academic Publishers.Norwell, MA, USA.200 p.

Zeichner, N., Berant J., and Dagan I. 2012. *Crowdsourcing Inference-Rule Evaluation.* in proc of ACL 2012 (short papers).