HAL
open science

Inference and Reconciliation in a Crowdsourced Lexical-Semantic Network

Manel Zarrouk, Mathieu Lafourcade, Alain Joubert

To cite this version:

Manel Zarrouk, Mathieu Lafourcade, Alain Joubert. Inference and Reconciliation in a Crowdsourced Lexical-Semantic Network. CICLING: International Conference on Intelligent Text Processing and Computational Linguistics, Mar 2013, Samos, Greece. lirmm-00816230

HAL Id: lirmm-00816230

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00816230

Submitted on 11 Jun 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Inference and Reconciliation in a Crowdsourced Lexical-Semantic Network

M. Zarrouk, M. Lafourcade, and A. Joubert

LIRMM, Montpellier
161, rue Ada - 34392 Montpellier Cedex - France
manel.zarrouk@lirmm.fr, lafourcade@lirmm.fr, alain.joubert@lirmm.fr

Abstract. *Lexical-semantic network construction and validation is a major issue in the NLP. No matter the construction strategies used, automatically inferring new relations from already existing ones is a way to improve the global quality of the resource by densifying the network. In this context, an inference engine has for purpose to formulate new conclusions (i.e. relations between terms) from already existing premises (also relations) on the network. In this paper we devise an inference engine for the JeuxDeMots lexical network which contains terms and typed relations between terms. In the JeuxDeMots project, the lexical network is constructed with the help of a* game with a purpose *and thousands of players. Polysemous terms may be refined in several senses (bank may be a bank>financial institution or a bank>river) but as the network is indefinitely under construction (in the context of a Never Ending Learning approach) some senses may be missing. The approach we propose is based on the triangulation method implementing semantic transitivity with a blocking mechanism for avoiding proposing dubious new relations. Inferred relations are proposed to contributors to be validated. In case of invalidation, a reconciliation strategy is undertaken to identify the cause of the wrong inference : an exception, an error in the premises or a transitivity confusion due to polysemy with the identification of the proper words senses at stake.*
Keywords: inference, reconciliation, lexical networks.

1 Introduction

Developing lexico-semantic network for NLP in one the the major issue of the field. Most of the existing resources have been constructed by hand, like for instance the famous WordNet. Of course some tools are generally designed for consistency checking, but nevertheless the task remains time consuming and costly. Fully automated approaches are generally limited to term coocurrences as extracting precise semantic relations between terms from text is really difficult. New approaches involving crowdsourcing are flowering in NLP especially with the advent of Amazon Mechanical Turk or in a broader scope Wikipedia and Wiktionary, to cite the most well known examples. WordNet ([1] and [2]) is such a lexical network based on synsets which can be roughly considered as concepts. [3] with EuroWordnet a multilingual version of WordNet and [4] for WOLF, a French version of WordNet, applied automated crossing of WordNet and other lexical resources with some manual checking. [5] constructed automatically BabelNet a large multilingual lexical network from the Wikipedia encyclopedia, but mostly with term coocurrences. Hownet [6] is another example of a large bilingual knowledge base (English and Chinese) containing semantic relations between word form, concepts and attributes. In Hownet there are much more different relations than in WordNet, although both projects started in the 80's as manual constructed by linguists and psychologists.

A highly lexicalized lexico-semantic network can contain concepts but also plain words (and multi-word expressions) as entry points (nodes) along with word senses. The idea itself of *word senses* in the lexicographic tradition may be debatable in the case of resources for semantic analysis, and we generally prefer to consider words usage. By *words usages* we mean refinement of a given word which is clearly identified by locutors but might not be always separated form other word usages of the same entry. A word usage put the emphasis on how and which context the term is actually used by locutors. A polysemic term has several usages that might differ substantially for word senses as classically defined. An given usage can also in turn have several refinements. For example, frigate can be a bird or a ship. A frigate>boat can be distinguished as a modern ship of an ancient vessel. In the context of a collaborative construction, such a lexical resource should be considered as being constantly under construction. For a polysemic term, some refinements might be just missing. As a general rule, we have no definite certitude

about the state of an entry. There is no way (unless by close 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 any similar resource in general) can be devised according to two strategies. First, as a contributive system 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 [7]) 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. Past experience shows that players and contributors complete the resource on terms that interest them. 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 application and notably semantic analysis. For example, contributors seldom indicate that a particular bird type can fly, as it is considered an obvious generality. Only notable facts which are not easily deductible are contributed. Well known exceptions are also generally contributed and take the form of a negative weight for the relation (for example, $\textit{fly} \xrightarrow{agent:-100} \textit{ostrich}$).

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

In this article, we first present the principles behind of lexical network construction with crowdsourcing and *games with a purpose* (also know as human-based computation game) and exemplified them with the JeuxDeMots project. Then, we present the outline of an *elicitation engine* based on an *inference engine* and a *reconciliation engine*. An experimentation is then reported on the performances of the system.

2 Lexical Network and crowdsourcing

There are many ways for building a lexical network considering some crucial factors as the quality of data, cost and time. Beside manual or automated strategies, contributive approaches are more and more popular as they are both cheap to set up and efficient in quality. More specifically, there is an increasing trend of using on-line GWAPs (game with a purpose [8]) method for feeding such resource.

The JDM lexical network is constructed through a set of on-line associative games. In these games, players are appealed to contribute on lexical and semantic relations between terms or verbal expressions which are presented in the network by the arcs interconnecting nodes in a graph. The informations in the JDM network are gathered by an unnegotiated crowd agreement (classical contributive systems rely on a negotiated crowd agreement).

2.1 JeuxDeMots: a GWAP for building a lexico-semantic network

JeuxDeMots[1] is a two player GWAP, launched in September 2007, that aims to build a large lexico-semantic network [9]. The network is composed of terms (as vertices) and typed relations (as links between vertices). It contains terms and possible refinements in a similar way to the WordNet synset [1]. There are more than 50 types for relations and relation occurrences are weighted. The weight of a relation is interpreted as a strength, but not directly as a probability of being valid neither a confidence level.

[1] http://jeuxdemots.org

When Player A begins a game, instructions concerning the type of lexical relation (synonyms, antonym, domain, etc.) are displayed, as well as a term T chosen from the database. Player A has a limited time to enter terms which, to his mind, correspond to term T and the lexical relation. A screenshot of the user interface during a game is shown in Figure 1 and of the outcome of the game in Figure 2.

The maximum number of terms a player can enter is limited, thus encouraging the player to think carefully about his choices. The same term T, along with the same instructions, are later given to another player, Player B, for whom the process is identical. To make the game more fun, the two players score points for words they both choose. Score calculation is explained in [10] and was designed to increase both precision and recall in the construction of the database. Answers given by both players are displayed, those common to both players are highlighted, as are their scores.

For a target term T, common answers from both players are inserted into the database. Answers given by only one of the two players are not, thus reducing noise (i.e. mistake from players, in this case) and the chances of database corruption (voluntary erroneous answers from a malicious player). The semantic network is therefore constructed by connecting terms by typed and weighted relations, validated by pairs of players. These relations are labelled according to the instructions given to the players and weighted according to the number of pairs of players who choose them. Initially, prior to putting the game online, the database was populated with nodes, however if a pair of players suggest a non-existing term, the new node is added to the database.

In the interest of quality and consistency, it was decided that the validation process would involve anonymous players playing together. A relation is considered valid if and only if it is given by at least one pair of players. This validation process is similar to that used by [7] for the indexing of images, and by [11] to collect common sense knowledge and [12] for knowledge extraction. As far as we know, this technique has never been used for building semantic networks. In NLP, other Web-based systems exist, such as Open Mind Word Expert [13], which aims at creating large sense-tagged corpora with the help of Web users, and SemKey [14] which makes use of WordNet and Wikipedia to disambiguate lexical forms referring to concepts, thus identifying semantic keywords.

More than 1200000 games of JeuxDeMots have been played since its launch in September 2007 corresponding of around 20000 hour of cumulated play (with 1 minute per game).

Fig. 1. Snapshot of an ongoing game.

Fig. 2. Snapshot of the result of the game, where three words were in common which are linked to the target term (Willy Wonka) in the JDM lexical network: chocolat(chocolate), chocolaterie(chocolate factory) and usine(factory).

2.2 Diko: a Contributive Tool for the JDM Network

Diko[2] is a web based tool for displaying the information contained in the JDM lexical network but also can be used as a contributive tool. The necessity to not rely only on the JeuxDeMots game for building the lexical network comes from the fact that many dans la figure 2 relation types of JDM are either difficult to grasp for a casual player or not very productive (not many terms can be associated). Furthermore, the need of such a contributive tool historically came from the players themselves as they wanted to become direct contributors of JDM.

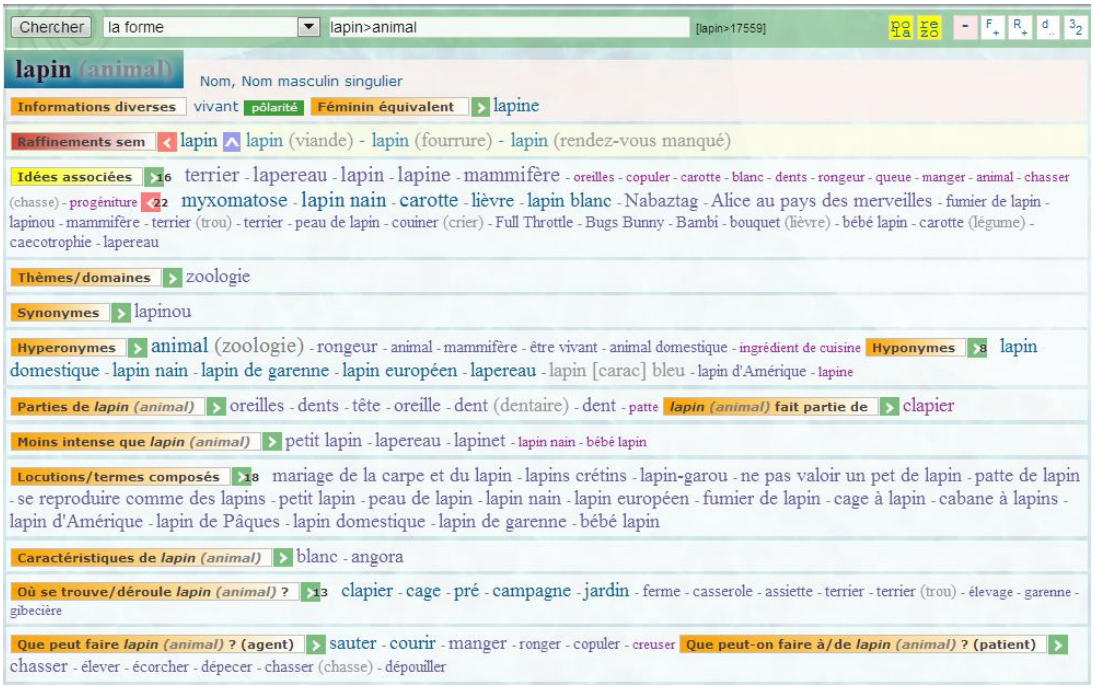

Fig. 3. Snapshot of a Diko screen for the term *lapin* (rabbit) as animal. In french, *lapin* can also refers to the fur, the meat, etc.

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

The principle of the contribution process is that a proposition made by a user will be voted pro or con by other users. When a given number of votes have been casted, an expert validator is notified and finally includes (or excludes) the proposed relation in the network. The expert can reject altogether a relation proposition or to include it with a negative weight if this is found to be relevant (for example, the relation $bird \xrightarrow{has-part:-100} fin$ may to be worthy as present in the network). Contributions can also be made by automatic processes to be scrutinized and voted by users. What we propose in this paper falls under this type of scenario or contributions / validations.

2.3 Some JDM Network Characteristics

At the moment of the writing of this article, JDM network contains 251358 terms and over 1500000 relations (amongst them 14658 being negative). More than 4500 terms have some refinements (various usages) for a total of around 15000 usages.

Although JDM has ontological relations, it is not an ontology per se with some clean and well-though 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, hyperonyme(cat) = {feline, mammal, living being, pet, vertebrate, ...}. In the previous hyperonyms set, we omitted weights for simplification, but in all generality, heavier terms are those felt by users as being the most relevant.

3 Inference and Reconciliation for an Elicitation Engine

We designed a system for augmenting the number of relations in the JDM lexical network having two main components: (a) an inference engine and (b) a reconciliator. The inference engine proposes relations as if it were 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 make some implicit knowledge of the user into explicit relations in the lexical network.

3.1 Inference Engine

The main ideas about inferences in our system are the following:

- inferring is for the engine to derive logical conclusions (under the form of relations between terms) from previously known premises, which are existing relations;
- candidate inferences may be logically blocked on the basis of the presence or absence of some other relations;
- candidate inferences can be filtered out on the basis of a strength evaluation;
- conclusions made by the inference engine are supposed to be true but may turn out to be *true*, *true but irrelevant* or *false* when proposed to a human validator.

In this paper, the type of inference we are working with, is based on the transitivity of the ontological relation *isa* (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 with C. The schema for the inference is as follows in Figure 4.

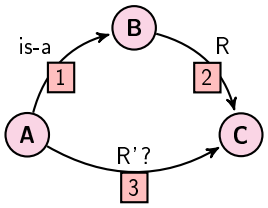

Fig. 4. Simple inference triangular schema applied to the transitivity of hyperonymy (*isa* relation). Relation (1) and (2) are the premises, and the relation (3) is the logical conclusion proposed in the lexical network and to be validated.

More formally, we can write: $\exists A \xrightarrow{is-a} B \quad \wedge \quad \exists B \xrightarrow{R} C \quad \Rightarrow \quad A \xrightarrow{R} C$

For example, $cat \xrightarrow{is-a} feline \quad \wedge \quad feline \xrightarrow{has-part} claw \quad \Rightarrow \quad cat \xrightarrow{has-part} claw$

Global processing

The inference engine is applied on terms having at least one hypernym (anyway the schema could not be applied otherwise). Let us consider a term T with a set of weighted hypernyms. From each hypernym, the inference engine deduces a set of inferences. Those inference sets are not disjoint in the general case, and the weight of an inference proposed in several sets is the incremental geometric mean of each occurrence. For example, as mentioned before, we have the following weighted hypernyms for cat: {felina, living being, mammal, pet, vertebrate, ...}.

The inference $cat \xrightarrow{has-parts} skeleton$ may comes from several hypernyms but strongly from *vertebrate*. The inference $cat \xrightarrow{location} house$ may come only from *pet*.

Logical filtering

Of course, this schema above is far too naive, especially considering the resource we are dealing with. In effect, B is possibly a polysemous term and ways to block inferences that are certainly wrong can be devised. If there are two distinct meanings of the term B that hold respectively the first and the second relation, then most probably the inference is wrong. This can be formalized (in a positive way) as follows:

$$\begin{gathered}
\exists A \xrightarrow{is-a} B \quad \wedge \quad \exists B \xrightarrow{R} C \\
\bigwedge \quad (\quad \exists B_i \xrightarrow{meaning-of} B \quad \wedge \quad \exists B_j \xrightarrow{meaning-of} B \quad) \\
\bigwedge \quad (\quad !\exists A \xrightarrow{is-a} B_i \quad \vee \quad !\exists B_j \xrightarrow{R} C \quad) \\
\Rightarrow \quad A \xrightarrow{R} C
\end{gathered}$$

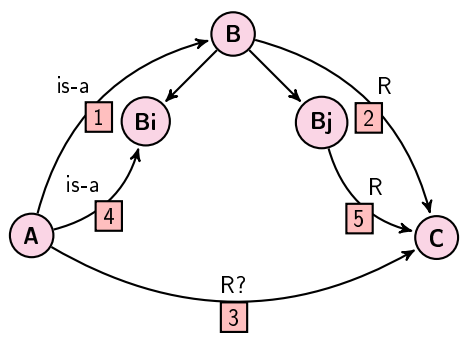

Fig. 5. Triangular inference schema with logical blocking based on the polysemy of the middle term B.

Moreover, if one of the premises is tagged as *true but irrelevant*, then the inference in blocked.

Statistical filtering

It is possible to evaluate a confidence level (on an open scale) for each produced inference, such a way that dubious inferences can be filtered out. The weight W of an inferred relation is the geometric mean of the weight of the premises (relations (1) and (2) in Figure 4). If one of the premises or both of them have a negative value, the weight is not considered as a number and the proposal is discarded. As the geometric mean is less tolerant to small values than the arithmetic mean, inferences not based on two rather certain relations (premises) are unlikely to pass.

$$\mathrm{W}(A \xrightarrow{R} C) = (\, \mathrm{W}(A \xrightarrow{is-a} B) * \mathrm{W}(B \xrightarrow{R} C)\,)^{1/2}$$

3.2 Reconciliation Engine

Inferred relations are presented to the validator to decide of their status: *true, true but irrelevant* or *false*. In case of invalidation, the reconciliator tries to diagnose the reasons: error as one of the premises (previously existing relations is false), exception or confusion due to polysemy (the inference has been made on a polysemous central term) and initiates a dialog with the user. The dialog should be succinct as to find the most informations with the fewest questions, without bothering the user but nevertheless trying to evaluate if the user is of good faith. To know in which order to proceed, the reconciliator determines if the weights of the premises are rather strong or weak. This confidence is done by comparing the relation weight against the threshold where the integrative value of the distribution of the relations separates the distribution in two equal parts (see Figure 6). For example, suppose we have:

term A: schoolboy
term B: human
term C: face
relation (1): $schoolboy \xrightarrow{is-a} human$

Figure 6 presents the distribution curve for all relations having a source term A (schoolboy) with a pike separating evenly the surface under the curve. The threshold is at 60 where the pike intersect the distribution curve.

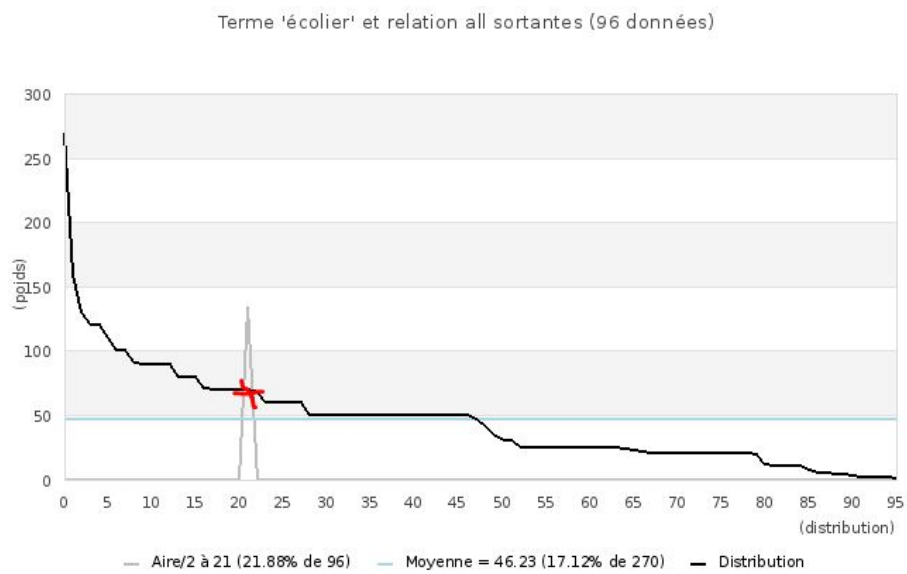

Fig. 6. Distribution of the outgoing relations for *Ãľcolier* (eng. schoolboy).

The confidence threshold for the relation (1) is the intersection between the distribution curve and the *area/2* curve.

- If W($A \xrightarrow{is-a} B$) >= confidence-threshold(A) $\Rightarrow$ $A \xrightarrow{is-a} B$ is a trusted relation;
- If W($A \xrightarrow{is-a} B$) < confidence-threshold(A) $\Rightarrow$ $A \xrightarrow{is-a} B$ is a dubious relation.

In the case we case both relations (1) and (2) as trusted, the reconciliator tries, by initiating a dialog with the validator, to check at first if the relation inferred is an exception. If not, it proceeds by checking if term B is polysemous and finally checks if it is an error case. We check the error case in the final step because the confidence level of relations (1) and (2) made them trusted. For example, suppose we have:

A:ostrich ; B:bird ; C:fly ; R:carac
(1): $ostrich \xrightarrow{is-a} bird$
(2): $bird \xrightarrow{carac} fly$
$\Rightarrow$ (3): $ostrich \xrightarrow{carac} fly$
In this case, it's true that an *ostrich is a bird* and that a *bird can fly*, but the inferred relation an *ostrich can fly* is a wrong one and it is considered as an exception.

In the case of having a dubious relation either for (1) and (2), the reconciliator suspect that it is an error case and this relation was the cause of a wrong inferences. So it asks the validator to confirm or to disprove it. In case of disapproval of one the relations we have an error. If not, proceed with checking if it's an exception case or a polysemy. For example, suppose we have:

A:kid ; B:human ; C:wing ; R:has-part
(1): $kid \xrightarrow{is-a} human$
(2): $human \xrightarrow{has-parts} wing$
$\Rightarrow$ (3): $kid \xrightarrow{has-parts} wing$
Obviously the relation $kid \xrightarrow{has-parts} wing$ is wrong and the relation $human \xrightarrow{has-parts} wing$ is the cause of the wrong inference.

Case of error in premises

In this case, suppose that relation (1) (in Figure 5) has a relatively low weight. The reconciliator asks the validator if the relation (1) is true .

- If the answer is negative, a negative weight is attributed to (1) and the reconciliation is completed; As such, this relation will not be used later on as premises on further inferences.
- If the answer is positive, ask if (2) is true and proceed as above if the answer is negative;
- Otherwise, move to the other cases (exception, polysemy).

Errors as exceptions

In the case we have two trusted relations, the reconciliator asks the validator if the inferred relation $A \xrightarrow{R} C$ is a kind of exception. If it is the case, the relation is stored in the lexical network with a negative weight along with a meta-information which indicates that it is an exception. Relations that are exceptions do not participate as premises.

Errors due to polysemy

In this case, if the middle term (B) presenting a polysemy is mentioned as polysemous in the network, the refinement terms B_1, B_2, ..., B_n are presented to the validator so he can choose the appropriate one. The validator can propose a new term as a refinement if he is not satisfied with the listed ones (inducing the creation of a new refinement). If there is no meta information that indicates the term is polysemous of the term, we ask first the validator if it is indeed the case. After this procedure, two new relations $A \xrightarrow{is-a} B_i$ and $B_j \xrightarrow{R} C$ will be included in the network with some positive value and the inference engine will use them later on (B_i and B_j being refinements of B choosen by the user).

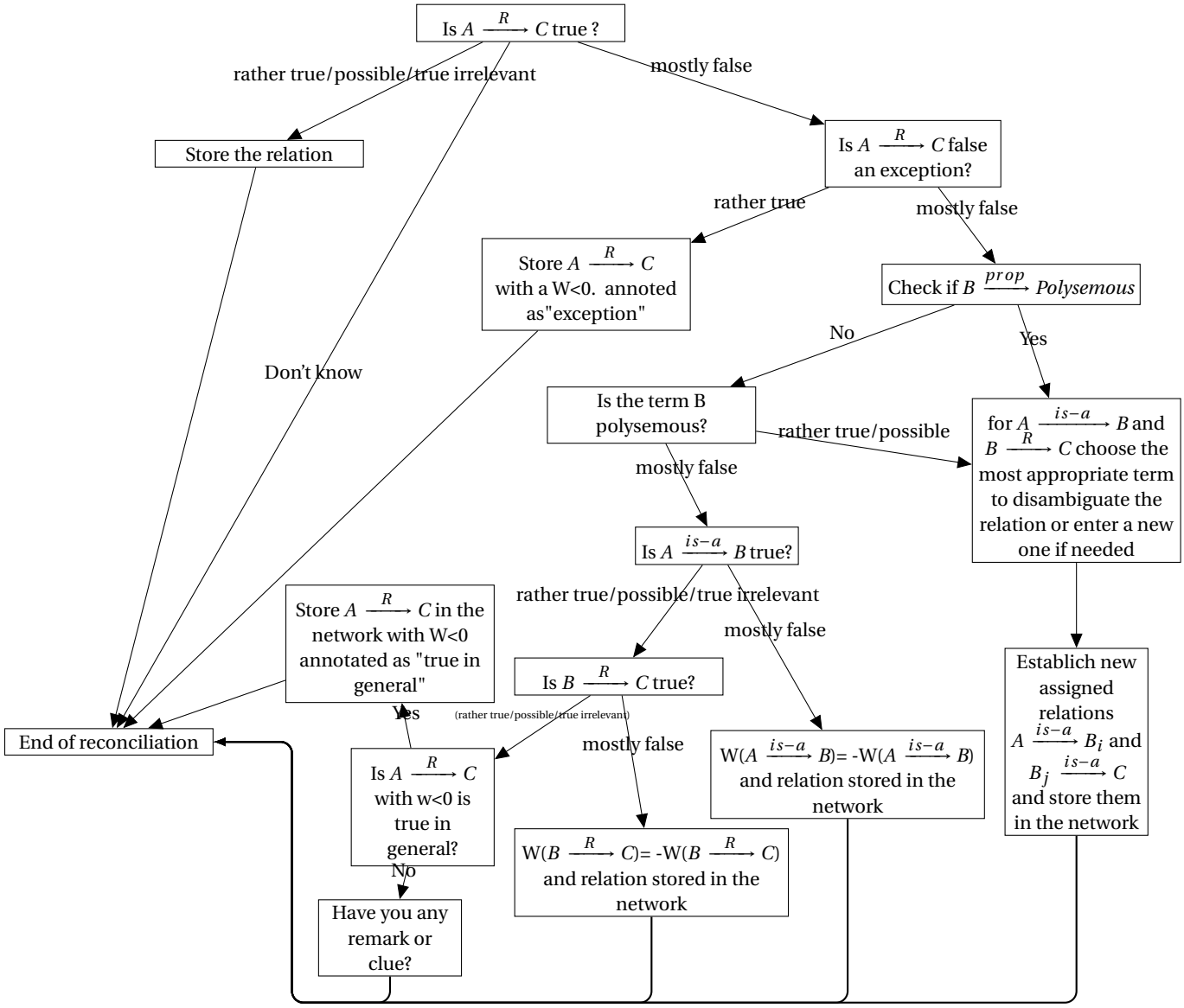

Fig. 7. Schema of the validation / reconciliation procedure.

4 Experimentation

We made an experiment with a unique run of the engine over the lexical network. The purpose is to access the production of the inference engine along with the blocking and filtering. Then from the set of supposedly valid inferred relations, we took a random sample of 300 propositions for each relation type and undertook the validation / reconciliation process. The experiment conducted is for evaluation purpose only, as actually the system is running iteratively along with contributors and games.

4.1 Unleashing the inference engine

We applied the inference engine on around 20000 randomly selected terms having at least one hypernym and thus produced 1484209 inferences (77089 more were blocked). The threshold for filtering was set to a weight of 25

(that is to say only inferences with a weight equal to or above 25 have been considered). This value is relevant as when a human contributor proposed relation is validated by an expert, it is introduced with a default weight of 25. In Figure 8, the distribution appears to be following a power law, which is not totally surprising as the relation distribution in the lexical network is by itself governed by such a distribution.

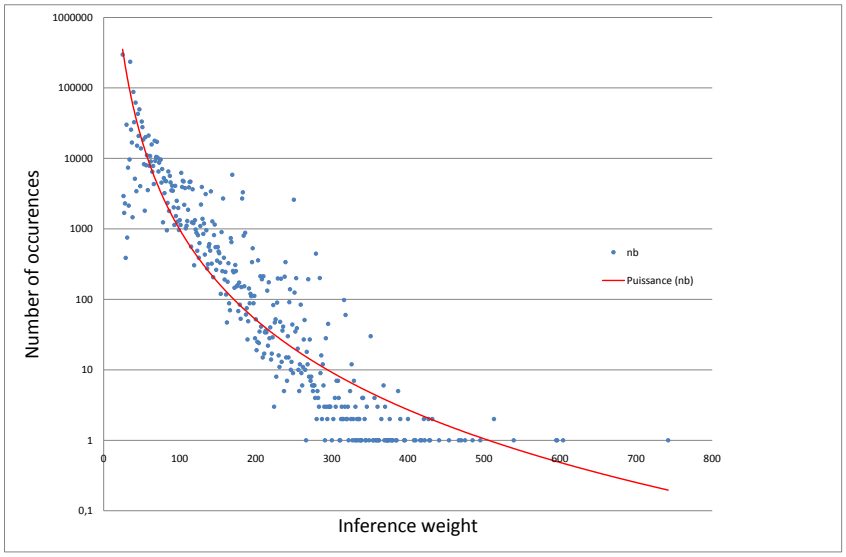

Fig. 8. Proposed inferences distribution according to weights. Weight bellow 25 are not displayed.

Relation types	**nb proposed**	**%**	**nb bloqued**	**%**	**nb flitered**	%
hypernym	91 037	6,13	4 034	5,23	53 586	26,32
has-parts	372 688	25,11	31 421	40,76	100 297	49,26
holonyme (inverse of has-parts)	108 191	7,28	17 944	23,27	26 818	13,17
typical location	271 717	18,30	11 502	14,92	14 174	6,96
caracteristics	203 095	13,68	2 647	3,43	6 576	3,23
agent-1 (what the subject can do)	198 359	13,36	9052	11,74	1122	0,55
instr-1 (the suject could be an instrument for what)	24 957	1,68	127	0,16	391	0,19
patient-1 (what can be done to the subject)	14 658	0,98	7	0,01	13	0,00
typical location-1 (what can be found in/on the subject)	145 159	9,78	129	0,17	206	0,10
lieu-action (what can be done in/on the subject)	50 035	3,379	91	0,12	132	0,06
object mater (mater/substance of the subject)	4 313	0,29	135	0,17	262	0,12
Total	**1 484 209**	100	**77 089**	100	**203 577**	100

Table 1. Status of the inferences proposed by the inference engine.

The table 1 presents the number of relations proposed by the inference engine. The different types for the second premise (the generic R relation in the inference triangulation) are variously productive. Of course, this is mainly due to the number of existing relations and the distribution of their type in the network. The productivity of a relation type is the ratio between the number of the proposed inferences and the number of occurrences of this relation type in the network.

The transitive inference on *isa* is the less productive which might seems surprising at first glance. In fact, the *isa* 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 agent semantic role (the *agent-1* relation) is by far the most productive, with 30 more propositions than what currently exists in the lexical network.

Relation type	**nb proposed**	**nb existing**	**productivity**
isa	91 037	91 799	99,16%
has-parts	37 2688	21 886	1702,86%
holonyme	108 191	13 124	824,37%
typical location	271 717	26 346	1031,34%
characteristics	203 095	24 180	839,92%
agent-1	198 359	6 820	2908,48%
instr-1	24 957	4 797	520,26%
patient-1	14 658	3 930	372,97%
typical location-1	145 159	8 835	1642,99%
location-action	50 035	4 559	1097,49%
object mater	4 313	3 097	139,26%

Table 2. Productivity of relation types when exploited by the inference engine.

4.2 Figures on Reconciliation

In table 3 is presented some evaluation of the status of the inferences proposed by the inference engine. 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 but are true) and false positives (inferences voted true but are false) are evaluated to less than $0,5\%$.

	% valid		**% error**		
Relation types	**relevant**	**not relevant**	**in premises**	**as exception**	**due to polysemy**
isa	76%	13%	2%	0%	9%
has-parts	65%	8%	4%	13%	10%
holonyme	57%	16%	2%	20%	5%
typical location	78%	12%	1%	4%	5%
characteristics	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%
typical location-1	72%	12%	2%	10%	6%
location-action	67%	25%	1%	4%	3%
object mater	60%	3%	7%	18%	12%

Table 3. Results of the validation / reconciliation according to inference types.

5 Conclusion

In this paper we presented some issues in building a lexico-semantic network with games and user contributions and inferring new relations from existing ones. Such a network is highly lexicalized and word usages are discovered incrementally along its construction. Errors are naturally present in the resources as they might come from

games played for difficult relations, but they are usually discovered by contributors for terms they are interested in. The same observation is generally done on what contributors contribute. To be able to enhance the network quality and coverage, we proposed an elicitation engine based on inferences and reconciliations. Inferences are here conducted on the basis of a simple triangulation based on the hypernymy transitivity, along a logical blocking and statistical filtering. A reconciliation process is conducted, in case the inferred relation is proven wrong, in order to identify the underlying cause. As global figures, we can conclude that inferred relations are true and relevant in about 78% of the cases and true but not relevant in 10% of the case. Overall wrong inferences is about 12% with at least one error in the premises of about 2%, exceptions of about 5% and polysemy confusion of about 5%. Beside being just a tool for increasing the number of relations in a lexical network, the elicitation engine is both an efficient error detector and polysemy identifier. The actions taken during the reconciliation forbid an inference proven wrong to be inferred again and again. Such an approach should be pushed forward with other type of inference schema and possibly with the evaluation of the distribution of the semantic classes of the terms on which inferences are conducted. Indeed some semantic classes like concrete objects or living beings may be substantially more productive for certain relation types than for example abstract nouns of processes or events. Anyway, such discrepancies of inference productivity between classes are worthy to investigate further.

References

1. Miller, G., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: Introduction to wordnet: an on-line lexical database. International Journal of Lexicography **3**(4) (1990) pp. 235–244
2. Fellbaum, C., Miller, G.: (eds) WordNet. The MIT Press (1998)
3. Vossen, P.: Eurowordnet: a multilingual database with lexical semantic networks. (1998) 200 p.
4. Sagot, B., Fier, D.: Construction d'un wordnet libre du français à partir de ressources multilingues. TALN 2008, Avignon, France, 2008. (2008) 12 p.
5. Navigli, R., Ponzetto, S.: 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 (2012) pp. 216–225
6. Dong, Z., Dong, Q.: HowNet and the Computation of Meaning. WorldScientific, London (2006)
7. von Ahn, L., Dabbish, L.: Designing games with a purpose. Communications of the ACM **51**(8) (2008) pp. 58–67
8. Thaler, S., Siorpaes, K., Simperl, E., Hofer, C.: A survey on games for knowledge acquisition. STI Technical Report, May 2011. (2011) 19 p.
9. Lafourcade, M.: Making people play for lexical acquisition. In Proc. SNLP 2007, 7th Symposium on Natural Language Processing. Pattaya, Thailande, 13-15 December 2007 (2007) 8 p.
10. Joubert, A., Lafourcade, M.: Jeuxdemots : un prototype ludique pour l'émergence de relations entre termes. In proc of JADT'2008, Ecole normale supérieure Lettres et sciences humaines , Lyon, France, 12-14 mars 2008 (2008) 8 p.
11. Lieberman, H., Smith, D.A., Teeters, A.: Common consensus: a web-based game for collecting commonsense goals. In Proc. of IUI, Hawaii. (2007) 12 p.
12. Siorpaes, K., Hepp, M.: Games with a purpose for the semantic web. In IEEE Intelligent Systems **23**(3) (2008) pp. 50–60
13. Mihalcea, R., Chklovski, T.: 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) (2003) 10 p.
14. Marchetti, A., Tesconi, M., Ronzano, F., Mosella, M., Minutoli, S.: Semkey: A semantic collaborative tagging system. in Procs of WWW2007, Banff, Canada (2007) 9 p.
15. Chamberlain, J., Poesio, M., Kruschwitz, U.: Phrase detectives: A web-based collaborative annotation game. In Proceedings of the International Conference on Semantic Systems (I-Semantics'08) (2008)
16. Chklovski, T., Gil, Y.: Improving the design of intelligent acquisition interfaces for collecting world knowledge from web contributors. In Proceedings of K-CAP'05 (2005) pp. 35–42
17. Feng, D., Besana, S., Zajac, R.: Acquiring high quality non-expert knowledge from on-demand workforce. In Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources, People's Web '09, Morristown, NJ, USA. Association for Computational Linguistics. (2009) pp. 51–56
18. Lafourcade, M., Joubert, A., Schwab, D., Zock, M.: Evaluation et consolidation d'un réseau lexical grâce à un assistant ludique pour le mot sur le bout de la langue. In proc of TALN'11, Montpellier, France, 27 juin-1er juillet 2011 (2011) pp. 295–306
19. Lenat, D.: Cyc: A large-scale investment in knowledge infrastructure. Communications of the ACM **38**(11) (1995) pp. 33–38
20. Ploux, S., Victorri, B.: Construction d'espaces sémantiques à l'aide de dictionnaires de synonymes. Traitement Automatique des Langues **39**(1) (1998) pp. 161–182
21. Zesch, T., Gurevych, I.: Wisdom of crowds versus wisdom of linguists measuring the semantic relatedness of words. Natural Language Engineering, Cambridge University Press. (2009) pp. 25–59