Ranking Agent Statements for Building Evolving Ontologies

Ronny Siebes & Frank van Harmelen

Department of Computer Science and Mathematics
Vrije Universiteit Amsterdam
[ronny,frankh]@cs.vu.nl

Abstract

In this paper a methodology is described for ranking information received by different agents, based on previous experience with them. These rankings again are used for asking the right questions to the right agents. In this way agents can build up a *reputation.* The methods in this paper are strongly influenced on human heuristics regarding the assignment of confidence ratings on humans. The methods provide a solution to current problems with ontologies: (1) handling contradicting and sloppy information, (2) efficient network use instead of broadcasting information and (3) dealing with ontological drift.

Introduction

This The WWW is a dynamic and heterogeneous area where information is provided by many different sources. It is often difficult to know the validity of information found at those different (sometimes anonymous) sites. Usually, human readers are sufficient capable judging the validity of information based on their experience with the provider and with the subject itself. In the case of contradicting information between different sources the user does not derive 'falsum' but believes the most probable one based on some (implicit) heuristics. In the relatively new area of the semantic web [Berners-Lee, 1999], machines are supposed to share knowledge via commonly agreed formal description of the concepts and their relations, i.e. ontologies. Unfortunately, these computers don't have these human heuristics, and therefore have no or limited means to 'judge' the correctness of the incoming information. This also means that it is difficult to determine which source is an expert on a certain area.
In a peer-to-peer environment, it is important that an agent determines a selection of agents that should be queried. This prevents that the network is flooded with broadcasted information. It also reduces the number of unwanted answers. Another problem with ontologies is the 'ontological drift', which means that information can be out-dated. We cannot assume that people maintain their ontologies properly, like they also lack maintaining websites. How frustrating can it be that you read a website, and after ten minutes reading you see that the information is out of date.

In this paper we try to get a grip on these three problems (handling sloppy or inconsistent information, peer selection and coping ontological drift) by a rating methodology already introduced by [Siebes, 2001]. This methodology means that every statement received by an agent is ranked with a certain value, the *confidence rating*. In this way agents build up a *reputation*. [Lethin et al.] used the notion of reputation already for summarizing the behavior of past transactions of peers. An example of reputations at work is the 'trust metric' at http://www.advogato.org, which is a portal for open source development work. The Advogato trust metric resists a scenario in which many people join the system with the express purpose of boosting each others' reputation scores [Levien]. Our confidence rating is based on previous experience with semantic information of the agent, and on statements of other agents. In this way, different 'truths' can exist in the ontology, however one truth can be more true than the other. Based on these ratings, the expertise of every individual source can be determined. If an agent wants to know something that is not present in his own ontology, it can determine a list of agents that have the highest probability to answer the query. At last, a devaluation method can be used on the ratings: every statement devaluates in time, if the statement is not strengthened on time, it will fade away and deleted out of the ontology.

This introduction showed three different problems in the ontology domain and introduces a rating methodology to solve them. Section two describes the different aspects of the rating methodology in an informal way. After this, section three shows that the methodology of section two solves the problems of section one.

Rating methodology

Before we describe the rating methodology some assumptions have to be made. We assume that we have a peer-to-peer agent network where every agent has its own ontology, can do queries and provide information (statements) to other agents. The ontologies are organized in an explicit is-a hierarchy (taxonomy). This implies that the class relationships from ontologies described in, for example, *description logic* [Fensel et al., 2001] have to be made explicit first. As already mentioned in the introduction, information provided by agents is not treated as an absolute truth, but as a possible truth.

The rating methodology involves the following aspects: (1) assigning confidence ratings on statements received by agents. (2) Updating these ratings when new information comes into the ontology. (3) Using these ratings to determine the appropriate expert for answering an incoming query that can't by answered by the agent itself. (4) Updating the assumptions made by determining the experts. (5) Applying an aging mechanism on ratings and removing statements that dropped below a certain value. We now describe these aspects in more detail:

Assigning confidence ratings

When a statement is provided by an unknown source it gets a (low) initial confidence rating. With 'unknown' we mean that the source never has provided any information to the receiver. Thus, when an agent *a* receives information from agent *b* and *b* is unknown to *a*, then the receiving agent has to decide if it includes the information of *b* into the ontology. If it wants to include this, and it has no clue about the reputation of *b*, it assigns an initial confidence to the information.

Updating confidence ratings

If other agents than the original sender repeat an existing statement, the statement gains higher confidence. This has the side effect that also the source of the existing statement becomes more an expert on the relevant concept and those close to it. To determine the 'closeness' of concepts towards each other we use the notion of *semantic distance* [Budanitsky]. How this works, will be explained in section three.

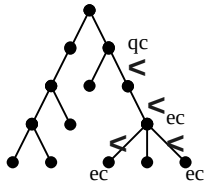
Determining the experts to be queried

If an agent receives a query (from a user, or another agent) and it cannot answer this query out of its ontology, it calculates a list of agents that have the highest probability to give a correct answer. Also here the semantic distance is of big importance: not only the reputation of the agents is important, but also the closeness of its expertise to the query.

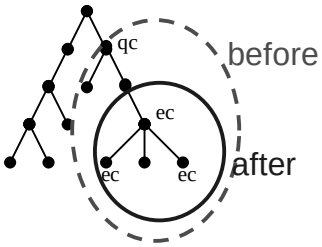
Updating assumptions made about determining the experts

If an agent, selected as an expert from the list above, answers with 'I don't know!', a wrong assumption is made about the expertise of the agent. Now we have two possibilities:

- increase the semantic distance between the concept in the query *qc* and the concepts of the expert *ec*.



- decrease the 'expert range' of the expert.



The choice between the first and the second option depends on the maturity of the ontology. If an is-a relation just was formed, the weight between the two classes is less fixed than in the case when relation survived already for a long time. Thus, the longer the is-a relations exist the more difficult it is to adapt the weight, and the more the expert range is of influence.

Aging mechanism to devaluate confidence ratings in time

An aging mechanism takes care of the obsolete statements. In time the confidence ratings devaluate, and if they drop below a certain treshold, the statement will be deleted. The only way to stay alive is that these statements are repeated now and then. In this way only the 'strong' statements will survive. This basic strategy can at least be extended with two enhancements:

1. *Statements that survived for a long time are less influenced by the aging mechanism as 'fresh' statements*.
   If a statement survived already for a long time, it is probably a statement that is always true, i.e. it is independent from time. For example, "Bach is-a musician", or water is wet.

2. *Statements higher in the tree are less influenced by the aging mechanism, then statements at the leaves***.**
   Statements high in the ontology tree are more general then those at the leaves. General statements are often so trivial that almost nobody makes them explicit, e.g. "a car has wheels" or "an animal is-a living_creature". To 'protect' these general statements, the aging

mechanism is not as tough then on the more specific statements.

3. *Distinguish different statement types in applying the aging mechanism.*
   There is a difference between absolute statements like "the speed of the HP880cxi printer is 12 ppm" and relative statements like "the speed of the HP880cxi printer is high". When a distinction is made in the ontology between those statements, we can use this knowledge to examine which statements are more influenced by time.

A positive side-affect of the aging mechanism, is its prevention of the inflate effect on increasing confidence ratings (experiments should fine-tune the mechanism). This side effect is needed because for now we only increase the ratings in case of confirmation of other agents, we don't decrease the ratings if agents say the opposite. The reason for this is the difficulty in finding opposite statements, e.g. the system should know that a person can't be a woman and a man, in order to determine that man(x) is the opposite of woman(x).

Conclusions

Now that we described the rating methodology, we will explain how it solves the three problems as described in the introduction. First, the rating methodology handles sloppy and contradicting information by adding ratings to statements. In case of two inconsistent statements the system can choose the most probable answer instead of deriving falsum. Second, an efficient peer selection method is introduced by selecting agents on their assumed expertise. Only ask those agents that have the highest probability to answer your query correctly. Third, the rating method deals with ontological drift by automatically degrading the ratings. If a statement is not strengthened in time, it will fade out of the network. In this way only the strong and the new statements reside in the ontology.

The rating method as here described contains a lot of parameters that have to be tuned in practice, therefore we started with implementing a system to do this. It will be a combination with the Protégé ontology editor [Noy et al., 2000] and the JADE agent platform [Bellifemine et al., 1999].

References

[Bellifemine et al., 1999]
Bellifemine F., Poggi, A., Rimassa, G. *JADE – A FIPA-compliant agent framework* CSELT internal technical report. Part of this report has been also published in Proceedings of PAAM'99, London, pag. 97-108, April 1999.

[Berners-Lee, 1999]
Berners-Lee T. *Weaving the Web*. Harper, San Francisco, 1999.

[Budanitsky et al., 2001]
Budanitsky A., Hirst G. *Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures*. Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics, Pittsburgh, June 2001.

[Fensel et al., 2001]
Fensel D., Horrocks I., Harmelen van F., McGuinness, D. and Patel-Schneider D.: *OIL: Ontology Infrastructure to Enable the Semantic Web*, IEEE Intelligent System, 16(2), 2001.

[Lethin et al., 2001]
Lethin, R. *Peer-to-peer: Harnessing the power of disruptive technologies*. O'Reilly ISBN 0-596-00110-x, March 2001.

[Levien]
Levien R. *Advogato's Trust Metric*. http://www.advogato.org/trust-metric.html

[Noy et al., 2000]
Noy N., Fergerson R., Musen M. *The knowledge model of Protege-2000: Combining interoperability and flexibility*. 2th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000), Juan-les-Pins, France, 2000.

[Siebes, 2001]
Siebes, R. *LARiSSA: A prototype implementation of evolving ontologies*. Masters' thesis, Vrije Universiteit Amsterdam, August 2001.