Towards a Framework for Trusting the Automated Learning of Social Ontologies

Konstantinos Kotis[1], Panos Alexopoulos[2], Andreas Papasalouros[1,3]

[1] University of the Aegean, Dept. of Information and Communication Systems Eng., Ai-Lab, Samos, Greece 83200
kotis@aegean.gr
[2] IMC Technologies, Athens, Greece
palexopoulos@imc.com.gr
[3] University of the Aegean, Dept. of Mathematics, Samos, Greece 83200
andpapas@aegean.gr

Abstract. Automatically learned social ontologies are products of *social fermentation* between users that belong in communities of common interests (CoI), in open, collaborative and communicative environments. In such a setting, *social fermentation* ensures automatic encapsulation of agreement and trust of the shared knowledge of participating stakeholders during an ontology learning process. The paper discusses key issues for trusting the automated learning of social ontologies from social data and furthermore it presents a framework that aims to capture the interlinking of agreement, trust and the learned domain conceptualizations that are extracted from such a type of data. The motivation behind this work is an effort towards supporting the design of new methods for learning *trusted* ontologies from social content i.e. methods that aim to learn not only the domain conceptualizations but also the degree that agents (software and human) may trust them or not.

1 Introduction

Web, Social Web and even Semantic Web content can be reused for the creation of semantic content, shaping information into ontologies. However a critical mass of useful semantic content is missing. Web users can only find few well-maintained and up-to-date domain ontologies and the amount of RDF data publicly available is limited compared to the size of the unstructured Web information. Only a small number of Web users, typically members of the Semantic Web community, build and publish ontologies. To assist and motivate humans in becoming part of the Semantic Web movement and contribute their knowledge and time to create or refine/enrich useful ontologies there is need to boost semantic content creation by providing Web users with a "starting point of assistance" i.e. automatically learned ontologies.

Traditionally, the learning of ontologies involves the identification of domain-specific conceptualizations that are extracted from text documents or other semi-structured information sources e.g. lexicons, thesauruses. Such learned ontologies do not utilize any available social data that may be related to the domain-specific data e.g. ownership details (contributor, annotator or end-user), tags or argumentation/dialogue items that have been used to comment, organize or disambiguate domain-specific information, querying information related to user clicks on retrieved information. Recently, the learning of ontologies has also involved social content that is mainly generated within Web 2.0 applications. Social content refers to various kinds of media content, publicly available, that are produced by Web users in a collaborative and communicative manner. Such content is associated to some social data that have been produced as a result of *social fermentation*. The most popular social data in Web 2.0 content is tags, which are (often) single words listed alphabetically and with a different font size or color (to capture its importance). Tags are usually hyperlinks that lead to a collection of items that are associated with. Such social data can be processed in an intelligent way towards shaping social content into ontologies. Since social data is produced as part of the social fermentation (tags are introduced in a collaborative and communicative manner), it can be argued that the learned ontologies that are produced from such a process encapsulate some degree of agreement and trust of the learned conceptualizations.

Social content generation (SCG) refers to a conversational, distributed mode of content generation, dissemination, and communication among communities of common interest (CoI). Social intelligence (SI) aims to derive actionable information from social content in context-rich application settings and to provide solution frameworks for applications that can benefit from the "wisdom of crowds" through the Web. Within this setting, a social ontology can be defined as: *an explicit, formal and commonly agreed representation of knowledge that is derived from both domain-specific and social data*. In the context of this chapter, the meaning of the term “social ontology” must be clearly distinguished from the meaning that is used in social sciences. A representative social-science definition is given by T. Lawson of the Cambridge Social Ontology Group[1]: “…the study of what is, or what exists, in the social domain; the study of social entities or social things; and the study of what all the social entities or things that are have in common”.

Formally, an ontology is considered to be a pair $O=(S, A)$, where S is the ontological signature describing the vocabulary (i.e. the terms that lexicalize concepts and relations between concepts) and A is a set of ontological axioms, restricting the intended interpretations of the terms included in the signature [3], [4]. In other words, A includes the formal definitions of concepts and relations that are lexicalized by natural language terms in S. In this paper, we extend such model by a social dimension (equal to *social semantics*) that is influenced by the definition of “Actor-Concept-Instance model of ontologies” [7] formulated as a generic abstract model of semantic-social networks. The extended model is build on an implicit realization of emergent semantics, i.e. meaning must be depended on a community of agents. According to the extended model, a social ontology can be considered a triple $O=(C, S, A)$, where C is the set of collaborating contributors that have participated in a *SCG* task, from which S and A have been derived using the *SI* found in C. The range however of C over both S

[1] T. Lawson, A Conception of Ontology, The Cambridge Social Ontology Group, 2004, http://www.csog.group.cam.ac.uk/A_Conception_of_Ontology.pdf

and A at the same time is not guaranteed, i.e. S may have been derived from C, but not A, which may have been automatically derived from external information sources such as a general ontology or lexicon e.g. from WordNet.

The automated learning of social ontologies can be seen as a two-dimensional problem. The first dimension concerns the automated creation of ontologies from content (social and domain-specific), and the second, the social dimension, concerns collaboration and communication aspects (the *social fermentation*) that are involved during the creation of the content. Since automation is also involved, and human agents do not participate in the conceptualizations' agreement process, a key issue here is the trust on the extracted ontological agreement from social data i.e. the certainty that contributors of shared conceptualizations about a specific domain have agreed on a common understanding about the domain and that such agreement is successfully extracted in an automated fashion from social data (e.g. in open Web agents' world where agents must trust each others conceptualizations about the domain of discourse in order to be able to collaborate within an agreed context). In terms of the "trust the content" problem, the paper follows the assumption that the content used as input in an ontology learning process is a social one (or content that is involved in social fermentation), thus it is, at least in some degree, agreed and trusted. Blogs, (Semantic) Wikis, Folksonomies and other more sophisticated Web 2.0 applications such as Yahoo!Answers or Fixya.com, provide reputation-based trust (use personal experience or the experiences of others, possibly combined, to make a trust decision about an entity) or voting mechanisms for their content. Other types of content such as Web users' query logs provide a kind of trusting their content, based on the "majority vote of user clicks" on Web search results.

To the best of our knowledge and from literature review [1], [8], [10], currently there is no mean to automatically discover and attach uncertainty values on automatically learned social ontologies' signature *(S),* axioms *(A)* and contributors *(C)*. This paper proposes a model that represents trust for an ontology of the form $O = \{C, S, A\}$. More specifically, trust is formed as a meta-ontology which represents meta-information related to each element of a social ontology i.e. classes, properties, instances, contributors. Such meta-information is related to social data (e.g. contributors details, voting information) that is in turn interlinked to the content represented in the domain ontology. The definition of $O = \{C, S, A\}$ is then extended, as shown in the paper, by introducing also trust representation.

The paper is structured as follows: section 2 presents the proposed framework for trusting social ontologies automatically learned by social content, section 3 reports on case studies for applying the proposed framework, and section 4 concludes the paper.

2 The proposed framework

2.1 Representing trust in social ontologies

This paper proposes a model that represents trust for an ontology of the form $O = \{C, S, A\}$. More specifically, it is formed as a meta-ontology which represents meta-information related to each element of a social ontology i.e. classes, properties, instances, contributors. Such meta-information is related to social data (e.g. contributors details, voting information) that is in turn somehow related to the content represented in the domain ontology. The definition of $O = \{C, S, A\}$ is then extended by introducing trust T for C, S and A such as $T = \{u, v_a, v_f\}$ where: u specifies the uncertainty value computed for a given instance of C, S or A, v_a specifies the number of votes that do not trust an instance of C, S or A, and v_f specifies the number of votes that do trust an instance of C, S or A. In other words, some trusted (with some degree of uncertainty) contributors C are trusting (with some degree of uncertainty) a particular class, property or instance (i.e. an instance of S ontological signature) or an axiom (i.e. an instance of A axioms) that is learned from *C's* contributed content. Although the computation of *u (uncertainty value)* reflects the trust in C, S or A within a social network of C contributors, v_a *and* v_f values are reflecting the absolute *number of agreement* among the members of C for a given member of C, S or A.

2.2 Integrate trust in HCOME-3O meta-ontologies framework

Ontologies are *evolving* and *shared* artefacts that are collaboratively and iteratively developed, evolved, evaluated and discussed within communities of common interest (CoI), shaping domain-specific information spaces. To enhance the potential of information spaces to be collaboratively engineered and shaped into ontologies within and between different communities, these artefacts must be escorted with *all* the necessary meta-information concerning the conceptualization they realize, implementation decisions and their evolution. In HCOME-3O framework [11], the integration of three (meta-)ontologies that provide information concerning the conceptualization and the development of domain ontologies, the atomic changes made by knowledge workers, the long-term evolutions and argumentations behind decisions taken during the lifecycle of an ontology, has been proposed (and evaluated via its utilization in later work). This involves ontology engineering tasks for a *domain* ontology and its versions (*domain knowledge*), i.e. editing, argumentation, exploiting and inspecting, during which meta-information is captured and recorded (*development ontologies*) either as information concerning a simple task or as information concerning the interlinking of tasks. This framework has been proposed in the context of HCOME collaborative engineering methodology [5].

Recently, HCOME methodology has been extended with ontology learning tasks [5] in order to capture knowledge that is automatically extracted from content and learned in the domain ontology. In such a new dimension of the methodological aspect of ontology engineering, agent agreement on automatically learned conceptualizations may be assisted by integrating representations of already computed uncertainty values in the following way: collaborating knowledge contributors consult uncertainty values of the learned conceptualizations and agree or disagree on the conceptualizations.

The integration of the proposed model into the HCOME-3O framework can be easily achieved by merging its semantics with the Administration meta-ontology [11], which mainly records instances of domain conceptualizations (classes, properties, individuals) and contributors of such conceptualizations, in the following way (Figure

1): a) add trust-related datatype properties ("uncertainty_value", "votes_against", "votes_for") of the trust model to the Administration meta-ontology, Administered_Item class), b) add object properties (has_superClass, has_Domain, has_Range, has_Type) to the corresponded ontology elements, extending the Administration meta-ontology, in order to facilitate the assignment of trust on (simple) axioms (*A)* also.

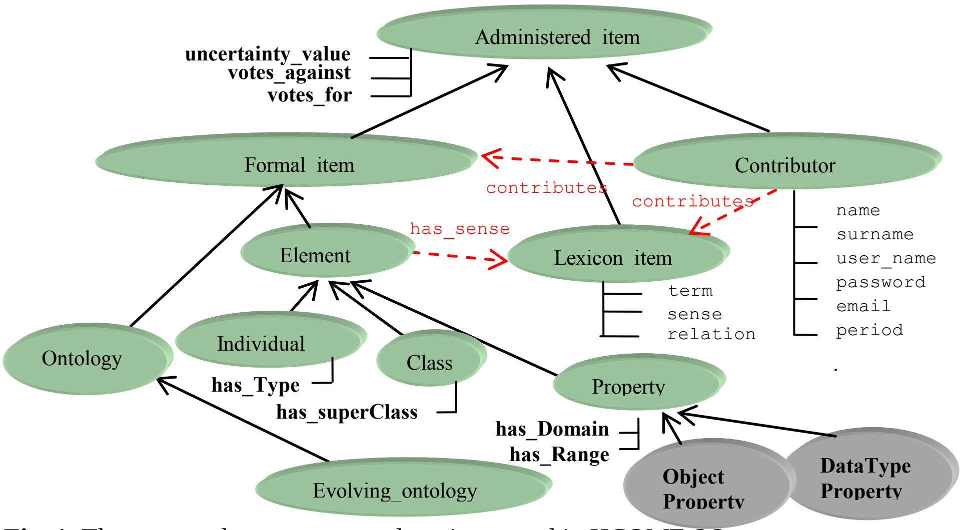

Fig. 1. The proposed trust meta-ontology integrated in HCOME-3O

A *Formal Item* (a Class, a Property or an Individual item of the domain ontology) is recorded in the meta-ontology as domain knowledge that is contributed by a specific Contributor. Trust-related properties are attached to formal items, represented with the dataType properties: *uncertainty_value*, *votes_against*, *votes_for*. Such properties are inherited to all formal items of the extracted signature *S* of the domain ontology. Furthermore, the trust-related properties are also attached (inherited via the Administered Item specification) to the Contributor. Such information is necessary in order to keep record of trusted (by others) people also. A similar conceptualization for trusting people is provided in Trust ontology of MindSwap (http://trust.mindswap.org/trustOnt.shtml) however it is not interlinked with trust-aware domain conceptualizations that these people may contribute. On the other hand, several other efforts have been lately presented (see related work section) for modeling trusted conceptualizations; however they are not interlinked with the trust-aware recording of their contributors. Having said that, since the Administration meta-ontology is part of a wider meta-information framework (that is HCOME-3O framework), the 'range' of the trust-related properties can be expanded to other meta-information also such as the changes that are recorded between each new version that contributors are developing, the argumentation dialogue items (arguments, issues, positions) that are recording during the collaborative evaluation/development of ontologies, etc. As a result of this effect, the model can provide answers to more complex and trust-elaborated queries such as '*give me all changes that were made be-*

tween version O_i and version O_{i+1} of the ontology O, by contributors with a trust T = {0.6, 4, 5}' or '*give me all suggesting positions (argumentation items) that were made by the community for classes of the domain ontology with a trust T = {0.6, 4, 5}* '.

For a detailed description of the Administration meta-ontology and its role in the HCOME-3O framework please refer to the related article [11].

Trust on ontological axioms (*A*) can be also extracted since they also comprise domain knowledge that can be discovered in social content. Trust on axioms however can be easily inferred from already trusted classes (upper level) and simple axioms (e.g. subsumption) of the learned ontology. Consider for instance axioms of a simple pre-specified topics' hierarchy (e.g. the one in Yahoo!Answer Web 2.0 application): If class A, class B and class C are trusted with value "1.0" and the axioms $A \sqsubseteq B$ and $B \sqsubseteq C$ are also trusted with values "1.0", then the inferred axiom $A \sqsubseteq C$ can be also trusted with value "1".

Concluding the paragraph, the definition of the social ontology can now be re-formulated as follows: $O = \{T, C, S, A\}$ where $T = \{u, v_a, v_f\}$ *is the trust function* $T: C \cup S \cup A \rightarrow [0,1] \times Z \times Z$ *for ontological signature S and axioms A that a set of collaborated contributors C participated in a task of social content generation (SCG) have derived based on their social intelligence (SI).*

2.3 Computation of uncertainty values

For a Web social application that uses a voting system to trust (or not) some content (e.g. in Yahoo!Answers application where users vote for or against a posted answer to a Yahoo!Answers community question), the uncertainty value *u* for this chunk of content can be computed using a simple formula $u = v_f - v_a$, i.e. the number of votes after subtracting the votes against (v_a) from the votes for (v_f). The vector of the voting values computed for some content, $U = (u_1, u_2 \ldots u_n)$ where *n* represents the number of content chunks that have been related to the voting system (e.g. in the Yahoo!Answers application *n* is the number of answers *a* posted for a question *q*), is then normalized to the interval *[0, 1]*. The normalization will work well if data is positive or zero. If data contains negative numbers, for example, *-1*, *3* and *4*, then the sum is 6. If it is normalized by the maximum value we get *-1/6*, *½*, and *2/3*. The sum of the three values is still *1* but now a negative number (*-1/6*) is part of the index. The following general solution may be however applied: Shift data by adding all numbers with the absolute of the most negative (minimum value of data*)* such that the most negative one will become zero and all other numbers become positive. Then data is normalized using any common normalization method for zero or positive numbers. For example, if data is *-1*, *3* and *4*, the most negative number is -1, thus we add all numbers with *+1* to become: *0*, *4*, *5* and then normalize it.

A general normalization solution for voting values in a social application is proposed in the following lines. Suppose we have a range or scale from A to B and we want to convert it to a scale of 1 to 10, where A maps to 1 and B maps to 10. Furthermore, we want to do this with a linear function, so that for example the point midway between *A* and *B* maps to halfway between 1 and 10, or 5.5. Then the following (linear) equation can be applied to any number *x* on the A-B scale:

$$y = 1 + (x - A) * (10 - 1) / (B - A). \quad (\mathbf{1})$$

Note that if x = A, this gives y = 1 + 0 = 1 as required, and if x = B, then:

$$y = 1 + (B - A) * (10 - 1) / (B - A) = 1 + 10 – 1 = 10, \quad (\mathbf{2})$$

as required. One can use this equation even if A > B. In our case, the scale will be 0.0 to 1.0 for every *x*, where $x \in \{u\}$.

2.4 Using the framework for automatically generating fuzzy ontologies

The fact that the trust model of the presented framework assigns uncertainty values to the elements of the ontology learned through the *social fermentation* process practically means that (most of) the knowledge represented by this ontology is uncertain. Typically, representation of uncertain knowledge is facilitated by *fuzzy ontologies*, namely ontologies that utilize the notions of *fuzzy set* and *fuzzy relation* [6] in order to suggest that certain pieces of knowledge should be considered as true at certain degrees.

As with traditional ontologies, the pure manual generation of a fuzzy ontology is a difficult and tedious task that requires the active involvement of domain experts, mainly for the task of assigning truth degrees to the ontology's elements. Since our framework provides a way for automatically performing this task, we claim that it may be as well used for the automatic generation of fuzzy ontologies. To show why this is the case we consider a formal definition of a fuzzy ontology, adapted from [12], in which the latter is a tuple $O_F = \{C, I, FR, FA, FLV, FVA\}$ where:

- *C* is a set of concepts (classes) and *I* is a set of individuals.
- *FR* is a set of fuzzy relations. Each fuzzy relation is a function $E^2 \rightarrow [0,1]$ where E is the union of *C* and *I*. Of particular importance are two fuzzy relations: the fuzzy subsumption relation between concepts and the fuzzy instantiation relation between concepts and instances.
- *FA* is a set of fuzzy attributes. Each fuzzy attribute is a function $I \rightarrow F(X)$, $F(X)$ being the set of all fuzzy sets in the universe of discourse *X*.
- FLV is a set of fuzzy linguistic variables. Each variable is a tuple $\{u, T, X, m\}$ in which *u* is the name of the variable, *T* is the set of linguistic terms of *u* that refer to a base variable whose values range over a universal set *X* and *m* is a semantic rule that assigns to each linguistic term a meaning in the form of a fuzzy set in *X*.
- *FVA* is a set of fuzzy valued attributes. Each fuzzy valued attribute is a function $I \rightarrow T$ where *T* is the set of the linguistic terms of a fuzzy linguistic variable.

Given this definition, generating a fuzzy ontology practically means assigning truth degrees to fuzzy relations and defining the meanings of fuzzy linguistic terms. As can be seen from figure 1, our framework supports the first from these two tasks through the assignment of uncertainty values to the *has_superclass* property of the *Class* item (fuzzy subsumption), the *has_type* property of the *Individual* item (fuzzy instantiation) and to the instances of the *Object* and *Datatype* property items (fuzzy relations and fuzzy attributes). The support of the second task, namely generation of linguistic term meanings, is left as future work as it requires an extension of the administration meta-ontology to the fuzzy realm.

Related work on the learning of fuzzy ontologies comprises methods that perform text mining in order to generate degrees for the fuzzy subsumption relation between concepts [6] [9] and for the fuzzy instantiation relation [6]. The work presented in this paper differs from these approaches in two ways. First of all it is more complete as it supports the automatic generation of any fuzzy relation, not only of the fuzzy subsumption and instantiation ones, as well as of fuzzy attributes. In addition to that, however, it follows a different perspective as uncertainty in the learned ontologies is not captured as an effect of a "good" or "bad" text mining technique but it is rather a result of the social fermentation process during the creation of social content (social data and domain-specific content). This does not only seem to be the right approach when referring to social ontology learning but it is in-line with the social dimension of the automatic learning ontology process.

Of course, the text-mining uncertainty dimension may be also of some importance when combined with a social one: A "gold" approach towards trusting automated learning of social ontologies can serve as a merger of both dimensions i.e. the text-mining and the social one. Intuitively, the average uncertainty of the two values can be considered the "gold" uncertainty value u_g of the formulae $T = \{u, v_a, v_f\}$ of our approach. However, more sophisticated formulas may be proposed, if based, for instance, on the work of learning trust decision strategies in agent-based reputation exchange networks [2], [8]. Assuming that a pessimistic strategy is followed [8], where agents do not trust each other unless there is a reason to do so, the uncertainty value of a text mining approach should be weighted more than the value of a social one.

3 Case Studies

In order to evaluate the proposed framework, it is necessary to develop and use ontology learning methods that learn social ontologies as a result of a *social fermentation* process. For this purpose we have re-used the in-house recently developed ontology learning method which utilizes (for input) mined domain-specific query logs of Web users community [5] and we are in the process of implementing an additional ontology learning method that utilizes Web 2.0 social content from Web Question/Answers applications such as Yahoo!Answers.

To apply the proposed trust framework on the Queries-to-Ontology learning method, an important assumption has been made since in such a context a voting mechanism is not present. The formula of $T = \{u, v_a, v_f\}$ is reduced to $T = \{u\}$ since in this case v_a and v_f can be considered of zero value. The computation of u for a Web query q is based on the reputation of the query in a particular context. Such reputation is reflected by the number of clicks *D_click(q)* on resulted documents D for a particular query q (reflecting that users' interests have been found in this query). Since this value can be considered as the reputation of a particular query, it can also be considered as the reputation of the learned conceptualizations from the particular query that a contributor *C* provided, i.e. the query-related signature (*S)* and axioms (*A)* of the learned ontology. Thus, the formula *O = {T, C, S, A}* is valid for this use case. Low *T* values will be returned for low *D_click(q)* values i.e. many Web users did not find search results to be much related to the query (they did not clicked on them). An additional step to this approach may be the analysis of history of queries: measuring the frequency of similar queries placed for the same context. This is left for future research.

Extending the work conducted using query logs as input to a social ontology learning process, a future direction is proposed in this paper, with the aim to trust the learning of social ontologies from Web 2.0 content. As a case study it was decided to apply the proposed framework on social content that is created by Yahoo! Answers community (an alternative is Fixya.com). Yahoo! Answers (http://answers.yahoo.com/) is a shared place where people collaborate and communicate by asking and answering questions on any topic. The aim of such a social fermentation is to build an open and commonly agreed knowledge base for the benefit of the community. Organized in topics (simple thematic category hierarchy), questions are posted by the users of the social network, expecting several answers that will eventually satisfy their knowledge acquisition needs. A voting for the best answer mechanism ensures that an agreed (by the majority) and trusted (by the number of "for" voters) answer is related to a question. Professional knowledge can also be shared within the community by *knowledge partners*. Such knowledge supplements the answers received from the community by answering questions in a specialized field, drawing on partners training, their professional experiences, and other appropriate resources. As a benefit, knowledge partners may mention their products or services, where relevant, in an answer, for advertisement reasons. Such a mutual benefit (for partners and community users) can guarantee a live social network that is difficult to "die" and at the same time it can guarantee the strong building of trust for the content that both stakeholders are sharing. The proposed method utilized the following inputs:

1) A question/answer document which contains the following information:
   a. the topic of the question (and the more general/specific categories of the topic hierarchy). Topics are pre-defined by Yahoo!Answers application
   b. user information: who posted the question, who posted an answer, who voted against or for
   c. the question and the associated answers in natural language: users can post a title and a comment for the question, and only comments for their answers
   d. the best answer and the votes for
   e. the votes for all other answers
   f. other related questions, resolved or open, on the same topic
2) WordNet lexicon. It will be used to enrich the ontology with additional semantics (entities, semantic relations, individuals)

The processing of the proposed social ontology learning method, integrated with the proposed trust framework, is outlined in the following steps (Figure 2):

- *Step-1*: The method learns the starting RDF triples from the types of the pre-defined hierarchy that the topic of the posted question is classified under.
- *Step-2*: The posted question (both title and comment) is analyzed using an NLP API (e.g. GATE[2]) in order to identify parts of speech (POS) and perform tokenization.
- *Step-3*: Since context is known (from Step-1) and some text analysis has been done (in Step-2), important terms can be identified and semantic relations between them can be recognized [5]. The following techniques can be used in combination:
    a. Hearst patterns
    b. Simple heuristic rules that utilize knowledge from the POS tagging.
- *Step-4*: Semantics are enriched using WordNet. Mapping of terms to WordNet senses is performed automatically using a statistical technique from Information Retrieval to compute latency of terms in term-document spaces (LSI method [6]).
- *Step-5*: Steps from Step-2 to Step-4 are repeated for the best (voted) posted answer. The ontology elements extracted from this step (classes, properties, instances) are assigned the uncertainty value 1.0 (representing the uncertainty of this element in respect to the community trust of the commonly agreed "best answer").
- *Step-6*: Steps from Step-2 to Step-4 are repeated for the rest posted answers. To keep the size of the learned ontology low (and to avoid noise) only important terms (most frequent terms) are introduced as classes of the learned ontology. The importance of terms is a threshold value that can be empirically set at '2'. However, in large sized answers (more than one paragraph of text) such value must be set higher. Other techniques should be also tested to avoid noise of large answers (e.g. to first locate important partitions of the text, applying n-grams analysis for instance, and then extract important terms from there). The ontology elements extracted from this step (classes, properties, instances) are assigned an uncertainty value (normalized) between the interval 0 and 0.9.
- *Step-7*: The generated RDF triples from steps Step-2 to Step-6 are transformed into a consistent OWL model. The development proposed is based on Jena API and Pellet.

[2] http://gate.ac.uk/

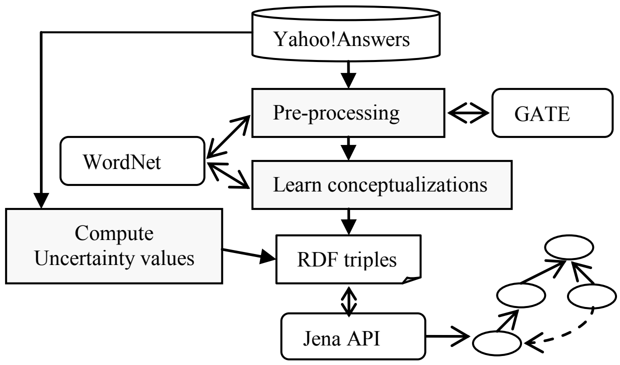

Fig. 2. The architecture of the proposed learning method

The output of the method is a learned ontology with uncertainty weights attached to its elements (classes, properties, instances). To respect the formula $O = \{T, C, S, A\}$, the learned ontology is recorded in the extended Administration meta-ontology of the HCOME-3O model, interlinking the trusted conceptualizations with trusted contributors. In this use case, the contributors are Yahoo!Answers voters, members of the Yahoo! community, for which trust values can be also computed using a) a point system that is provided by the application in order to represent the reputation in the community, and b) their experience in the community (time of registration).

The voting mechanisms integrated in Yahoo!Answers as well as in other Web 2.0 related applications (e.g. Fixya.com) provide social data that is able to relate some content i.e. a posted answer, to some other content, i.e. to a posted question, and to their contributors. Such interlinking can be interpreted as agreement or disagreement on users' opinion and eventually as a trust value of the shared knowledge that is encapsulated in the most agreed opinion (best voted answer). Trusted more or less, the related-to-a-topic knowledge is shaped into a domain ontology where each element is eventually associated with an uncertainty value that is computed directly from the social data associated with the represented content. Professional knowledge can also be shared within the community by *knowledge partners*. Such knowledge supplements the answers received from the community. Since this kind of knowledge is contributed by experts, it can be considered as highly trusted. Furthermore, the mutual benefit of knowledge partners and community users (advertisement and expertise knowledge contribution) plays a key role to "truth telling" when it comes to partners' answers in community users' posts. This can guarantee a live social network with strong roots of trust for the content that all stakeholders are sharing. Relatively to ontology learning from query logs method, the proposed ontology learning method can be trusted in a higher degree since its social data is both directly and indirectly associated with the content represented in the ontology.

4 Conclusions

This paper presents an effort towards devising a framework for trusting automatically learned social ontologies as part of a *social fermentation* between users that belong in communities of common interests (CoI), in open, collaborative and communicative environments. The paper discusses key issues towards this goal and focuses on the presentation of the model that interlinks agreement and trust with the learned domain conceptualizations that are extracted from social data of Web applications. The reported work contributes in the design of new ontology learning methods in a Web of trusted conceptualizations and their contributors. More specifically, the proposed framework can be used for consultation during the design of ontology learning from social data methods that need to automatically learn not only the domain conceptualizations but also the degree that agents trust these conceptualizations (and their contributors) or not.

References

1. Artz, D., Yolanda Gil: A survey of trust in computer science and the Semantic Web. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 5 58--71 (2007)
2. Fullam K., Barber S.: Learning Trust Strategies in Reputation Exchange Networks, AAMAS'06, May 8-12, Hakodate, Hokkaido, Japan (2007)
3. Kalfoglou, Y., M. Schorlemmer: Ontology mapping: the state of the art. The Knowledge Engineering Review 18 1--31 (2003)
4. Kotis, K., G. Vouros, K. Stergiou: Towards Automatic Merging of Domain Ontologies: The HCONE-merge approach. Journal of Web Semantics , 4 60--79 (2006)
5. Kotis, K., Papasalouros A.: Learning useful kick-off ontologies from Query Logs: HCOME revised. 4th International Conference on Complex, Intelligent and Software Intensive Systems (2010)
6. Lau, R., Li Y and Xu Y.: Mining Fuzzy Domain Ontology from Textual Databases, IEEE/WIC/ACM International Conference on Web Intelligence (2007)
7. Mika, P. Ontologies are us: A unified model of social networks and semantics. Journal of Web Semantics, 5 5--15 (2007)
8. O'Hara, K., H. Alani, Y. Kalfoglou, N. Shadbolt: Trust strategies for the semantic web. Proceedings of Workshop on Trust, Security, and Reputation on the SemanticWeb, 3rd International Semantic Web Conference (2004)
9. Tho, Q.T., Hui, S.C., Fong, A.C.M., Cao, T.H.: Automatic Fuzzy Ontology Generation for Semantic Web. IEEE Transactions on Knowledge and Data Engineering, 18 842--856 (2006)
10. Tang, J., Leung, H., Luo, Q., Chen D., Gong, J.: Towards Ontology Learning from Folksonomies, JCAI-09, Pasadena, California, USA, 11th - 17th July (2009)
11. Vouros, G., Kotis K, Chalkiopoulos C, Lelli N.: The HCOME-3O Framework for Supporting the Collaborative Engineering of Evolving Ontologies. ESOE 2007 International Workshop on Emergent Semantics and Ontology Evolution, ISWC (2007)
12. Alexopoulos, P., Wallace, M., Kafentzis, K., Askounis D.: Utilizing Imprecise Knowledge in Ontology-based CBR Systems through Fuzzy Algebra. International Journal of Fuzzy Systems, Special Issue on Fuzzy Approaches for Ontology Applications and Adaptive Web Services, in press.