Dynamic Iterative Ontology Learning

Christopher Brewster José Iria Ziqi Zhang

Fabio Ciravegna Louise Guthrie Yorick Wilks

Department of Computer Science,
University of Sheffield, Sheffield, S1 4DP, UK
Initial.LastName@dcs.shef.ac.uk

Abstract

We present a novel approach to ontology learning which takes an iterative view of knowledge acquisition for ontologies. Current systems view the ontology learning process as single pipeline with one or more specific inputs and a single static output. Our approach is founded on three open-ended resources: a set of texts, a set of learning patterns and a set of ontological triples, and the system seeks to maintain these in equilibrium. As events occur which disturb this equilibrium, actions are triggered to re-establish a balance between the resources. We present a gold standard based evaluation of the final output of the system, the results of which are significantly better that those found in previous work.

Keywords

Ontology Learning, Ontology Evaluation, Semantic Web, Knowledge Acquisition, Knowledge Management

1 Introduction

Ontologies have become the most commonly accepted form of knowledge representation in a wide range of fields including the Semantic Web, e-Science, e-Business, and Knowledge Management. The importance of reducing the manual effort involved in building them is undisputed. The core challenge in order to reduce this 'knowledge acquisition bottleneck' lies in learning ontologies from natural language texts, because, although there are other approaches (e.g. [8] where ontologies are learnt from software APIs), they have much more limited application.

An underlying assumption in many approaches to Ontology Learning (OL) from text is that the text corpus input to OL is, *a priori*, both representative of the domain in question and sufficient to build the ontology. This is, in our view, inadequate. For example, [13] write, regarding their system: "the main restriction [...] is that the quality of the corpus must be very high, namely, the sentences must be accurate and abundant enough to include most of the important relationships to be extracted". In our view, requiring an exhaustive manual selection of the input texts defeats the very purpose of automating the ontology building process. Closely related to this is what we consider to be the other fundamental failure of current approaches, which is to view the ontology learning process as single pipeline with one or more specific inputs and a single static output.

In this paper, we present a novel approach to ontology learning which takes an iterative view of knowledge acquisition for ontologies. Our approach is founded on three open-ended resources: a set of texts, a set of learning patterns and a set of ontological triples, and the system seeks to maintain these in equilibrium. Each resource may have additional items added to it: further documents can be added from an external repository or the web, further extraction patterns can be learnt, and further knowledge triples can be extracted from the documents. As events occur which disturb this equilibrium, actions are triggered that aim to re-establish the balance between the resources. The main advantage of our approach is its more accurate model of the way knowledge is continuously changing, uncertain and dependant on the evidence currently available and the confidence we have in that evidence.

This paper is organised as follows: In Section 2, we present some of the requirements of concerning OL, followed by a description of the system in Section 3. In Section 4, we describe the evaluation and this is followed by a discussion of the experiments. Related Work is presented in Section 6, followed by a Conclusion.

2 Requirements Analysis

A successful ontology learning method must take into account certain observations about knowledge and language: **1.** Knowledge is not monolithic, monotonic or universally agreed. It is uncertain, revisable, contradictory and differs from person to person. **2.** Knowledge changes continuously over time and so will be revised and re-interpreted continuously. **3.** Ontologies are inherently incomplete models of domains, but need to be maximally "fit for purpose.". **4.** Texts assume the reader has a certain amount of background knowledge. The great majority of ontological knowledge is in this background knowledge, and not in the text. **5.** While it is easy to establish that some relationship exists between two terms, explicit defining contexts are relatively rare in texts.

The set of resources an OL system manipulates -

the text, the ontology, and the extraction patterns - are intrinsically incomplete at any given stage.The best possible input specification of the task for the OL system to perform is given by a seed ontology, a seed corpus and a seed pattern set. It also follows from the above that it is not possible to completely specify the task *a priori* - the ontology engineer should able to intervene by pointing out correct/incorrect or relevant/irrelevant ontological concepts and documents, as the process runs, effectively delimiting the domain incrementally through examples. Given the dynamic nature of knowledge, our approach should allow for the continuous development of knowledge over time, as more resources are added. Therefore, another fundamental requirement of our approach is for the OL process to be viewed as an incremental rather than an one-off process - the output of one system run can be used as input to another run in order to refine the knowledge. Finally, the data sparsity problem necessitates the use of multiple sources of information.

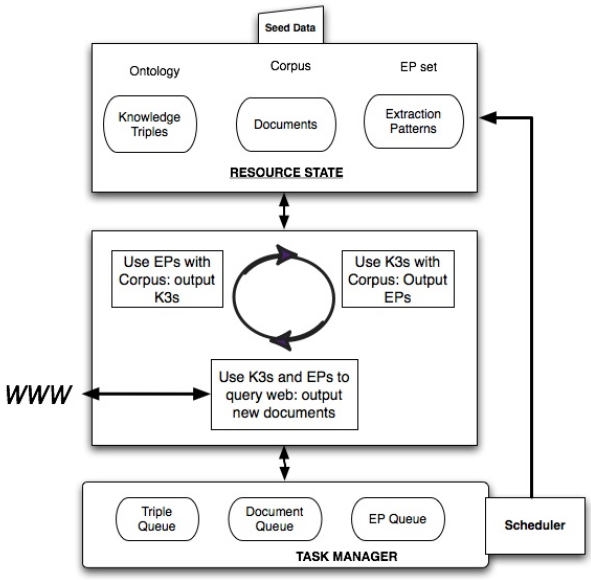

Fig. 1: *Overview of the system*

3 The Abraxas Approach

Our incremental, weakly-supervised approach views OL as a process involving three resources: the corpus of texts, the extraction patterns set (conceived as a set of lexico-syntactic textual patterns), and the ontology (conceived as a set of RDF triples). The goal is to extend existing resources in terms of one another, always seeking a consistent overall state which we will name *equilibrium*. Our method allows equally creating an ontology given an input corpus, extending a corpus given an input ontology or deriving a set of extraction patterns given an input ontology and an input corpus. The overall system can be seen in Figure 1.

The initial input to the processserves both as a specification of the task and as seed data for a bootstrapping cycle where, at each iteration, a decision is made on which new candidate concept, relation, pattern or document to add to the domain. Such a decision is modelled via three unsupervised classification tasks that capture the interdependence between the resources: one classifies the suitability of a pattern to extract ontological concepts and relations in the documents; another classifies the suitability of ontological concepts and relations to generate patterns from the documents; and another classifies the suitability of a document to give support to patterns and ontological concepts. The notion of "suitability" is formalised by assigning the relationship of any resource to the domain a confidence value, which we will denominate "resource confidence" (RC).

3.1 The Resource Confidence Measure

The Resource Confidence measure (RC) measures the confidence that the system has in a given resource i.e. item of knowledge, extraction pattern or document. The RC value for a knowledge triple reflects how confident the system is that it is a correct piece of knowledge, for an extraction pattern that the pattern will extract accurate pieces of knowledge, and for documents that the document provides valid knowledge triples. System added resources, whether documents, knowledge triples or extraction patterns are assumed to have varying degrees of confidence which is a function of the success or suitability of a given resource in deriving the corresponding other resource. Thus for each resource set, confidence for any resource item is defined in terms of the other resource sets. This means that for any given resource, there is a corresponding set of resource pairs with which it interacts.

The formulae for calculating the RC of any given resource are designed so that a) a single measure combines the effect of the other types of resources; b) the greater the sum of the confidence/RC values of the other resource pairs a given resource is associated with, the greater is the RC of that resource; c) the measure should take into account resource pairs not covered.

For example, for a given knowledge triple t_i, we aim to combine in one single measure the effect of both extraction patterns which extract the triple, and the documents that the triple is extracted from. An extraction pattern-document pair is defined as the instance of an extraction pattern applied to a given document. The measure favours knowledge triples that are the outcome of many extraction pattern-document pairs (instances) and favours triples that cover extraction pattern-document pairs with a high confidence.

Let O be the set of co-occurences of resource pairs - in this case as we are calculating the RC of a triple, the relevant resource pairs are document-extraction pattern pairs. We can conveniently represent this as a triple e.g. $o_1 = \{d_2, p_1, t_2\}$ which means that occurrence o_1 refers to document d_2 which has a match with EP p_1 to extract knowledge triple t_2. In the following formulas, d_r and d_w are restricted to the specific document in question, while d_p and d_n sum over all documents. Let d_r and d_w be the number of correct and incorrect documents in the set of document-extraction pattern pairs which output the triple t_i, and d_p and d_n be the number of positive and negative documents in the set of document-extraction pattern pairs which output *all* triples.

$$d_r = \sum_{o \in O_t} RC(d_o) \quad (1)$$

$$d_w = \sum_{o \in O_t} (1 - RC(d_o)) \quad (2)$$

$$d_p = \sum_{o \in O} RC(d_o) \quad (3)$$

$$d_n = \sum_{o \in O} 1 - RC(d_o) \quad (4)$$

Similar functions can be defined analogously for p_r, p_w, p_p and p_n. For further details and examples cf. [1].

r and w are defined in terms of the quantities defined in formulae 1 to 4 and the analogous formulae for p_r, p_w, p_p and p_n. r is defined as shown in Eq. (5), where d_r - d_n are defined as above. The quantity r trivially combines the contribution of both extraction patterns and documents by summing d_r and p_r. A refinement of the quantity, for ranking purposes, is obtained by adding the quotients, which favour triples that cover a greater number of positives, but less and less so as the number of negatives not covered increases.

$$r = d_r + \frac{(d_n - d_w)}{((d_n - d_w) + (d_p - d_r)) + 1} + p_r + \frac{(p_n - p_w)}{((p_n - p_w) + (p_p - p_r))} \quad (5)$$

$$w = d_w + \frac{(d_p - d_r)}{((d_p - d_r) + (d_n - d_w)) + 1} + p_w + \frac{(p_p - p_r)}{((p_p - p_r) + (p_n - p_w))} \quad (6)$$

The quantity w is the symmetric of the formula for r as shown in Eq. (6). Then the Resource Confidence (RC) for a given Knowledge Triple (for example t_i) is defined as shown in Eq. (7) which is merely the classic precision measure adapted for our purposes.

$$RC(t_i) = \frac{r}{(r + w)} \quad (7)$$

User-provided RC scores work as seeds and/or feedback to the system thereby optionally guiding the system as it runs. Extraction patterns are currently represented as described in [4]. The incompleteness of the corpus is tackled by iterative augmentation using the web or any other institutional repository as a corpus. Corpus augmentation in our approach consists of a set of methods that aim to incrementally add new documents to the corpus, such that documents with higher relevance to the domain are added first. Stopping criteria are established by setting a threshold on the lowest acceptable RC for each resource type, or by setting a threshold on the maximum number of iterations, without any new candidate resources for each resource type being obtained.

```
Corpus C = {d} a set of documents
Extraction Pattern Set P = {p} a set of extraction
patterns
Ontology O = {t} a set of knowledge triples

{
1.  State (seed) data (C, P, O)
2.  Candidates queues set to empty (C', P', O')
3a.  Apply P and term recognition (using a Noun Phrase
chunker) to discover triples in C;
3b.  Apply pattern induction to discover p in C;
3c.  Download more texts by applying O with Ps;

4.  Score discovered resources with RC;
5.  Place each discovered resource into corresponding
candidate queue (CC, CEP, CT);
6.  Pop the resource with the highest RC from the
candidate queues and add it to state (C, EP, T);
7.  Apply rationalisation;
8.  Re-score resources in C',P',O' and C, P, O;
9a.  If a triple t has been added, instantiate P with t to
query the web and download more texts using the triple t;
9b.  If an extraction pattern p has been added, apply p
over state C to discover new triples;
9c.  If a document d has been added, apply P and term
recognition over the text to discover triples;
10.  Go to Step 2;
}
```

Table 1: *The Bootstrapping Algorithm*

3.2 Bootstrapping Algorithm

The bootstrapping algorithm is shown in Table 1. Bootstrapping starts with the user providing some seed data (1,2). Initial processing includes applying the extraction patterns to the seed corpus to extract any available knowledge triples (3a), and learning new extraction patterns (3b). If the seed corpus is small, additional texts are obtained from the WWW by querying a search engine using the seed ontology and extraction patterns and added to the seed corpus (3c). A small corpus defines the domain weakly, in which case the RC scores would not correctly reflect the relevance of a resource to the domain.

The knowledge resources extracted by the initial processing are scored by applying the RC formula (4), and placed in the three resource queues (5). The queues contain candidate resources, sorted based on their RC in descending order, to be processed in following iterations.

The Scheduler component (see Figure 1) determines the following steps (6), in which the bootstrapping process polls the queues, and adds one resource to the system state at a time. Different schedulers implement different measures to determine which type of resources to be polled. In the experiment reported in this paper, the scheduler compares the RCs of the topmost resource in each queue, and adds the one with the highest RC to the state. Other measures which, for example, reflect how users intervene with the system and whether the user wants to supervise ontology learning, or corpus building, or pattern induction are also implemented, but not used in our current experiment.

Once a resource is added to the state, the bootstrapping applies rationalisation (7) and re-scores the state and candidate resources (8). Rationalisation rearranges the ontology so as to remove redundancy and make the ontology more coherent.

Following the addition of the resource, a new learning iteration is triggered (9). The system then contin-

ues cycling through the stages described above, (see Table 1) and iterates until stopping criteria are met.

4 Evaluation

Ontology evaluation is challenging topic in itself because knowledge cannot easily be enumerated, catalogued or defined *a priori* so as to allow for some sort of comparison to be made with the output of ontology tools. Various proposals have been made in the literature and an evaluation by Gold Standard (GS) was chosen in our case. For that purpose, we created a domain-specific hand-crafted ontology reflecting common sense knowledge about animals, containing 186 concepts up to 3 relations deep[1]. In order to compare the GS ontology with the computer generated one, we chose to follow the methodology proposed by Dellschaft and Staab [3]. The following metrics are thus used: Lexical Precision (LP) and Recall (LR) measure the coverage of terms in the GS by the output ontology; Taxonomic Precision (TP) and Recall (TR) aims to identify a set of characteristic features for each term in the output ontology and compare them with those of the corresponding term in the GS; Taxonomic F-measure (TF) is the harmonic mean of TP and TR, while Overall F-measure (TF') is the harmonic mean of TP, TR, LP and LR.

As a seed corpus we used a set of 40 texts from Wikipedia all entries concerning animals which were included in the GS ontology. All were entries for commonly known animals such as *hedgehog, lion, kangaroo, ostrich, lizard*, amounting to a little over 8000 words. Note there is a substantial gap between the number of animals initially covered in the articles and the number present in the GS ontology. The articles were pre-processed to remove the markup present in the originals.

A series of experiments were conducted, each time varying the seed knowledge input to the *Abraxas* system (in this paper we only present the one experiment, where Corpus = 40 Wikipedia texts, and Ontology = {dog ISA animal} - fuller details may be found in [1]). In all cases we used as a stopping criterion the Explicit Knowledge Gap (EKG) measure described in [6, 1]. This is a measure of the extent to which the ontology and the corpus are in equilibrium in the sense of the corpus providing explicit evidence for the items in the ontology. EKG is defined in Eq. 8 where E is the set of pairs of terms whose ontological relationship is explicit, Π is the set of pairs of terms in the corpus that are known to have some kind of ontological relationship on distributional grounds. The systems seeks to minimise EKG but in practice we use an empirically chosen threshold.

$$EKG = |E \cap \Pi| \tag{8}$$

We used the same set of 6 extraction patterns, shown in Table 2, which previous research had shown to have good precision [1]. Pattern learning was disabled in order to separate concerns - we intended to isolate the ontology learning process from the influence of pattern learning in these experiments, making results more comparable with those of the literature. For the same reasons, the system was tested in a completely unsupervised manner.

[1] Publicly available from `http://nlp.shef.ac.uk/abraxas/`

NP(pl) such as NP*	NP(sg) is a kind of NP(sg)
NP(sg) or other NP(pl)	NP(sg) is a type of NP(sg)
NP(pl) and other NP(pl)	NP(pl) or other NP(pl)

Table 2: *Extraction patterns used: NP = noun phrase, sg = singular, pl = plural.*

LP	0.40	LR	0.48
TP	0.95	TR	0.70
TF	0.81	TF'	0.60

Table 3: *Results obtained for experiment 1.*

Comparison with Gold Standard Our initial experiment was with Case 1, running over approximately 500 iterations. The final results are shown in Table 3. Both the TF and TF' obtained are significantly better than equivalent results in the literature, which often achieve maximum scores around [0.3] for both precision and recall [2].

Learning Curves Figure 2 shows how the results vary over the number of iterations. We can see here that LR steadily increases reflecting the growing size of the ontology and correspondingly its overlap with the GS. In contrast, LP is in constant flux but with a tendency to decrease. TP varies between set limits of [1.0 - 0.84] indicating that concepts are generally inserted correctly into the hierarchy. TR is also a measure in considerable flux and manual analysis of the different output ontologies show that sudden insertion of parent nodes (e.g. *mammal* at iteration 9) make a substantial difference which gradually stabilises over further iterations. Over long numbers of iterations, this flux in TR seems to become less likely. We also observe a steady increase TF' in parallel with the increase in LR indicating that the system is doing better as it increases its coverage of the lexical layer of the GS ontology.

5 Discussion

The low LP and LR do not accurately reflect the real quality of the generated ontology. LP has a tendency to decrease because the system is using the Web as a corpus, so it will inevitably include items absent from the GS. On the other hand, manual inspection

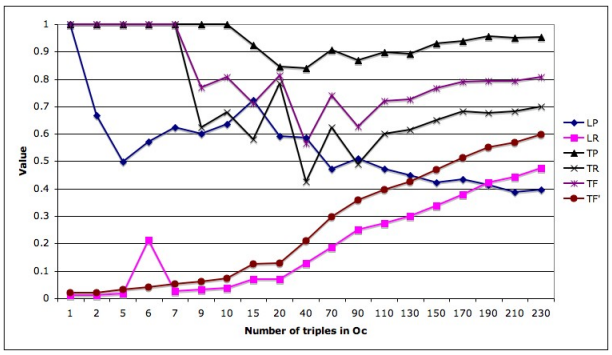

Fig. 2: *Evaluation measures (LP, LR, TP, etc.) plotted against the sequentially produced ontologies from the iterative process.*

of the ontology showed that in 230 triples, there were 225 concepts of which only 14 could be clearly seen to belong to another domain (flour, book, farmer, plant etc.), and another 10 were borderline (predatory bird, domestic dog, wild rabbit, large mammal, small mammal, wild fowl, etc.). So a manual evaluation would suggest 201 correct terms or [0.89] precision. The gradually falling LP presents a challenge for ontology learning and may either need a different approach to evaluating this element or a need for techniques which focus the ontology more effectively.

The flux shown in the graph presented in Figure 2 in the early stages shows that in principle as more data is added to the system the output becomes more stable and consistent. The general tendency is for the measures to move upwards indicating a gradual but steady improvement over the progression of the iterations. These results are as was hoped and reflect the capacity of the system to adapt as the data added to the system changes the confidence values for individual items of knowledge. The high F measures for the system show that our approach has fundamental validity.

Given the high quality of the output of this approach the question arises whether this is really what is needed. Is this type of ontology too focussed and does it just succeed algorithmically to re-create the well-known tennis problem [11]? This can only be answered by further experimentation and evaluation, varying the parameters of the approach.

6 Related Work

For an over view of research in OL, please consult [9]. More extensive descriptions of related work can be found in [6, 1].

The original inspiration for using lexico-syntactic patterns is [5] and developed by many other authors since. A number of authors have worked on ways to build ontologies accessing resources beyond the original corpus, e.g. [2] experiment with using data from WordNet, the Web (in general) and the counts provided by Google; [10] introduced an approach for automatically acquiring hypernyms and hyponyms for any given term using search engines. The bootstrapping learning approach inspiration from [14], [12] and [4]. Combining the use of the Web as a corpus and the bootstrapping approach, Etzioni et al. have created the KnowItAll system to collect factual information for a given domain, and provided one module that learns taxonomic relations [7].

7 Conclusion

In this paper, we have presented an iterative dynamic and adaptive system for ontology learning. The system is designed to achieve a balance between three open ended resources, a corpus, an ontology and a set of extraction patterns. We have described the key principles that lead to the system design and the key aspects of the system architecture and shown in our evaluation that the system is able to generate domain specific ontologies of good quality (TF' = [0.5 - 0.6]).

There are a number of objectives in our future work. First we plan to perform experiments to identify where the methodology fails especially concerning abstract concepts which are absent from the text collection. Secondly, we plan to fully evaluate the influence of pattern learning in the overall ontology learning process with a series of new experiments. Finally, we plan to investigate the application of our approach in important domains such as biomedical texts.

References

[1] C. Brewster. *Mind the Gap: Bridging from Text to Ontological Knowledge*. PhD thesis, Department of Computer Science, University of Sheffield, 2007.

[2] P. Cimiano, A. Pivk, L. Schmidt-Thieme, and S. Staab. Learning taxonomic relations from heterogeneous sources of evidence. In *Ontology Learning from Text: Methods, Evaluation and Applications*, Frontiers in Artificial Intelligence. IOS Press, 2005.

[3] K. Dellschaft and S. Staab. On how to perform a gold standard based evaluation of ontology learning. In I. F. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, editors, *International Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, pages 228–241. Springer, 2006.

[4] O. Etzioni, M. J. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web an experimental study. *Artif. Intell.*, 165(1):91–134, 2005.

[5] M. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING 92), Nantes, France, July 1992*, 1992.

[6] J. Iria, C. Brewster, F. Ciravegna, and Y. Wilks. An incremental tri-partite approach to ontology learning. In *Proceedings of the Language Resources and Evaluation Conference (LREC-06), 22-28 May*, Genoa, Italy, 2006.

[7] A.-M. Popescu, A. Yates, and O. Etzioni. Class Extraction from the World Wide Web. In *Proceedings of the AAAI-04 Workshop on Adaptive Text Extraction and Mining (ATEM-04)*, San Jose, CA, July 2004.

[8] M. Sabou. From software APIs to web service ontologies a semi-automatic extraction method. In S. A. McIlraith, D. Plexousakis, and F. van Harmelen, editors, *International Semantic Web Conference*, volume 3298 of *Lecture Notes in Computer Science*, pages 410–424. Springer, 2004.

[9] M. Shamsfard and A. A. Barforoush. The state of the art in ontology learning: a framework for comparison. *The Knowledge Engineering Review*, 18(04):293–316, 2004.

[10] R. Sombatsrisomboon, Y. Matsuo, and M. Ishizuka. Aquisition of hypernyms and hyponyms from the WWW. In *Proc. of 2nd International Workshop on Active Mining (AM2003)*, pages 7–13, Maebashi, Japan, 2003.

[11] M. Stevenson. Combining disambiguation techniques to enrich an ontology. In *Proceedings of the Fifteenth European Conference on Artificial Intelligence (ECAI-02) workshop on Machine Learning and Natural Language Processing for Ontology Engineering, Lyon, France*, 2002.

[12] M. Thelen and E. Riloff. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 214–221, Philadelphia, PA, July 2002.

[13] S.-H. Wu and W.-L. Hsu. SOAT a semi-automatic domain ontology acquisition tool from chinese corpus. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002),*. Academia Sinica, ACLCLP, and National Tsing Hua University, Taiwan, Morgan Kaufmann, August 2002.

[14] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen. Automatic acquisition of domain knowledge for information extraction. In *COLING 2000, 18th International Conference on Computational Linguistics, Proceedings of the Conference, 2 Volumes, July 31 - August 4, 2000, Universität des Saarlandes, Saarbrücken, Germany*, pages 940–946. Morgan Kaufmann, 2000.