An Incremental Tri-Partite Approach To Ontology Learning

José Iria, Christopher Brewster, Fabio Ciravegna and Yorick Wilks

Natural Language Processing Group, Department of Computer Science,
University of Sheffield, Sheffield, U.K.
{J.Iria, C.Brewster, F.Ciravegna, Y.Wilks}@dcs.shef.ac.uk

Abstract

In this paper we present a new approach to ontology learning. Its basis lies in a dynamic and iterative view of knowledge acquisition for ontologies. The Abraxas approach is founded on three resources, a set of texts, a set of learning patterns and a set of ontological triples, each of which must remain in equilibrium. As events occur which disturb this equilibrium various actions are triggered to re-establish a balance between the resources. Such events include acquisition of a further text from external resources such as the Web or the addition of ontological triples to the ontology. We develop the concept of a knowledge gap between the coverage of an ontology and the corpus of texts as a measure triggering actions. We present an overview of the algorithm and its functionalities.

1. Introduction

There is a substantial and growing body of research on Ontology Learning (OL) in large part due to the importance of ontologies in application areas such as the Semantic Web, Agents and Knowledge Management. One of the main challenge lies in learning ontologies from texts, because, although there are other approaches (e.g. Sabou (2004)), they have more much more limited application and do not fundamentally overcome the 'knowledge acquisition bottleneck' which has plagued AI since its inception. It is this bottleneck which makes ontology learning a foundational challenge.

In this paper we present a novel approach to ontology learning which depends for its success on the correct degree of abstraction over the learning process. Our approach has been inspired by a number of different stimuli. One has been the Quinean view of knowledge which sees knowledge as a force field in a state of dynamic tension. Quine believed human knowledge "impinges on experience only along the edges". Statements in the field of knowledge that came into conflict with experience would result in a re-evaluation or re-adjustment of the statements or logical laws we have constructed. He considers that no particular experience is linked to particular statements of belief except indirectly "through considerations of equilibrium" affecting the whole field of knowledge or science. Thus statements can be held true "in the face of recalcitrant experience by pleading hallucination" (Quine, 1951). This notion of equilibrium and the cumulative effect of experience, or in our case textual events, is fundamental to our approach. Each encounter in a text, where one term is juxtaposed with another, is to be treated as evidence for an ontological relationship. However, it only the cumulative accretion of sufficient evidence from sources in which the system can have confidence which can be interpreted as ontological facts i.e. knowledge. Thus in this kind of approach there are no absolutes, only degrees of confidence in the knowledge acquired so far.

Another significant impetus has been previous research which has led to a much more subtle understanding of the relationship between a text or collection of texts and the knowledge that they 'contain.' Existing approaches to Ontology Learning from text tend to limit the scope of their research to methods by which a domain ontology can be built from a corpus of texts. The underlying assumption in most such approaches is that the corpus input to OL is, *a priori*, both representative of the domain in question and sufficient to build the ontology. For example, (Wu and Hsu, 2002) writes, 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. Furthermore, previous research (Brewster et al., 2003) has shown a discrepancy between the number of ontological concepts and the number of explicit ontological relations (relating those concepts) that can be identified in any domain-specific corpora. This 'knowledge gap' problem essentially occurs due to the nature of the texts – they lack so-called 'background knowledge', i.e. definitional statements which are explicit enough to allow the automatic extraction of the relevant ontological knowledge. A text is an act of knowledge maintenance not knowledge creation in that its intent is based on the assumption the reader will share a considerable amount of of 'common knowledge' in order to be able to process the text at all. Thus a given text will only modify or 'maintain' the knowledge assumed by that text. It is exactly the assumed knowledge which OL wishes to capture, and thus only by having a fuller understanding of the nature of texts that an appropriate methodology can be constructed for finding the knowledge available.

In the context of the Abraxas project, we have developed an approach to OL in which three language resources, namely ontology, corpus and lexico-syntatic patterns, are treated equally as incomplete resources to be augmented and refined by the OL process. The process consists of an interplay between three unsupervised classification tasks working over the resources in an iterative fashion. In this paper, we give a general overview of the Abraxas approach and then focus on its corpus augmentation facet, which tackles the aforementioned knowledge gap problem. In Section 2, we present an overview of the Abraxas approach, in Section 3 we present the idea of a knowledge gap between the ontology and the corpus and in Section 4, we provide an overview of the algorithm. A review of the relevant literature is followed by a conclusion.

2. The Abraxas Approach

The Abraxas approach is founded on viewing ontology learning as a process involving three resources: the corpus of texts, the set of learning patterns, and the ontology (conceived as a set of triples). Each may be seen in an abstract sense as a set of entities with specific structural relations. The corpus is composed of texts which may (or may not) have a structure or set of characteristics reflecting the relationship between them e.g. they may all come from one organisation, or one subject domain. The learning patterns are conceived as a set of lexico-syntactic patterns or more abstractly as a set of functions between certain textual phenomena and an ontological relationship of greater or lesser specificity. The ontology is also a set of knowledge triples (term - relation - term, or rather domain - predicate - range) whose structure may grow more and more complex as more items of knowledge are collected.

The goal in Abraxas is to create or extend existing language resources in terms of one another, with optional and minimal supervision by the user. The methodology allows, for instance, creating an ontology given an input corpus, extending a corpus given an input ontology or deriving a set of lexico-syntatic patterns given an input ontology and an input corpus. The initial input to the process, whether ontology, corpus, patterns or combinations thereof, serves both as a specification of the domain of interest 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 within a probabilistic framework, in which the relationship of any resource to the domain is assigned a confidence level. Thus, ontology, corpus and patterns grow from the maximum probability core (the initial input) to the lower probability fringes as the process iterates. Initially newly added elements have an initial low probability but as the data they provide is further confirmed this confidence in them increases (or decreases accordingly).

Stopping criteria are established by setting a threshold on the lowest acceptable probability 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. The premise of Abraxas, that input resources are intrinsically incomplete, has had a defining impact on the overall methodology. Firstly, since the specification of the domain of interest is given by seed ontology, corpus and patterns, it follows that it is not possible to completely specify the task *a priori*. In fact, given an incomplete domain specification, most OL approaches either favour correctness of the acquired knowledge to the detriment of coverage of the domain or completeness of coverage to the detriment of correctness. In Abraxas, the OL process is viewed as an incremental rather than a one-off process. The ontology engineer is able to (but not required to) intervene by pointing out correct/incorrect or relevant/irrelevant ontological concepts, documents and so on, as the process runs, effectively delimiting the domain incrementally through examples. Secondly, incompleteness of the corpus is tackled by iterative augmentation using the web as a corpus. Corpus augmentation in Abraxas 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.

3. The Knowledge Gap and Corpus Management

The standard approach to ontology learning views it essentially as a pipeline with a set of domain specific texts as input and a set of ontological triple as output. This may be augmented by accessing further resources such as WordNet or Google frequency counts (Cimiano et al., 2005). However, this remains an essentially linear process and as such conceives of knowledge in a monolithic manner (Brewster and O'Hara, 2006). In Abraxas, we view the knowledge acquisition process i.e. OL as iterative and cyclical.

We define initially a core corpus which may be either a given set of domain texts or a set of texts retrieved using a seed ontology to provide the query terms. Whether retrieved from the Web or from a specific text collection is immaterial. At an abstract level this core corpus contains a certain amount of knowledge, or more accurately, in the light of Brewster et al. (2003), assume background knowledge which is the relevant ontology we wish to build. The *knowledge gap* (KG) is the difference between an existing ontology and a given corpus of texts[1]. The knowledge gap is measured by identifying the key terms in the corpus and comparing these with the concept labels or terms in the ontology. Thus if O_T is the set of terms in the ontology and C_T is the set of terms in the corpus, then KG is define in Eq. 1.

$$KG = 1 - \frac{O_T}{C_T} \tag{1}$$

Clearly at the beginning of the ontology learning process, KG will initially be either 1 (maximal) or very close to 1, indicating a large 'gap' between the knowledge present in the ontology and that which needs to be represent which is latent in the corpus. As the ontology learning process progresses, the objective is to minimise KG as much as possible while realising that for reasons of Zipfs' law this is an asymptote.

There are a wide variety of methods for identifying the salient or key terms in a corpus of texts (e.g. Maynard and Ananiadou (2000) or Ahmad (1995)) but the real challenge is to automatically learn the ontological relationship between terms (Brewster and Wilks, 2004). It also relatively un-contentious to use distributional methods to identify that

[1] This is closely related to the notion of 'fit' we have proposed elsewhere (Brewster et al., 2004)

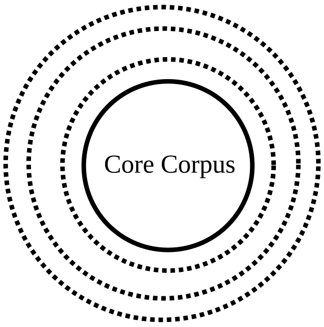

Figure 1: Iterative Expansion of the Core Corpus

there *exists* an ontological relationship between two terms. In Brewster et al. (2003), we defined *explicit knowledge* as textual environments where ontological knowledge is expressed in a lexico-syntactic pattern of the type identified by Hearst (1992). In such a case, the ontological relationship (domain - predicate - range) is automatically extractable. However, for any given corpus, a minority of the terms will occur as explicit knowledge and thus can be automatically added to the set of ontological knowledge. The difference between the set of terms whose ontological relationships are known and those which need to be added to the ontology (because they are key terms in the corpus) and whose ontological relationship is unknown, we will term the *Explicit Knowledge Gap* (EKG). The absence of explicit knowledge may be between two unaccounted terms or between an unaccounted term and a term already assigned to the ontology set. Thus $R_T \subset C_T$ is the set of pairs of terms in the corpus which are known to have some kind of ontological relationship on distributional grounds, and $E_T \subset O_T$ is the set of pairs of terms whose ontological relationship is explicit. E_T is a subset of the set of terms in the set of ontological knowledge because if the relationship is explicit then this knowledge can be added to the ontology. Thus EKG is defined analogously to Eq. 1 as follows:

$$EKG = 1 - \frac{E_T}{R_T} \tag{2}$$

While EKG will never be 0, in a similar manner to KG one objective of the Abraxas system is to minimise this Explicit Knowledge Gap. The seed or core corpus will inevitably have relatively high KG and EKG measures. The expansion of the corpus occurs in order to reduce the two respective knowledge gaps and consequently learn the ontology. As this is an iterative process we can conceive of the expanding corpus like ripples in a pool i.e. a set of concentric circles cf. Figure 1.

4. Description of the Overall Algorithm

In this section we describe the algorithm used in Abraxas, at a level of abstraction adequate to understand the mechanics of the ontology learning process. We start by introducing the main data structures used by the algorithm:

Corpus and Document - A Corpus is a collection of Document objects. Each Document object holds a graph representation of the linguistic features of the document text as well as the likelihood of the document belonging to the domain.

Wrapper and Pattern - A Wrapper is a collection of learning patterns or Pattern objects i.e. lexico-syntactic patterns. Each Pattern object holds a graph walk over the graph representation of the document and likelihood of the pattern belonging to the domain.

Ontology and Triple - An Ontology is a collection of Triple objects. Each Triple object holds the domain, predicate and range of the relation (to use RDFS terminology) and the likelihood of the triple belonging to the subject domain.

All three resources types - Document, Pattern and Triple - have an associated probabilistic value which measures the likelihood of the resource belonging to the domain. Values of 1 and 0 are typically employed to provide the system with examples or counter-examples for learning, respectively. A value of 0,5 is to be interpreted as “nothing can be said about this resource”. During the learning process, the system constantly updates these likelihood values.

Additionally, the algorithm makes use of the following auxiliary data structures:

Information Focus - a collection of settings that define the way Abraxas uses the web as a corpus.

User Profile - a collection of settings that define the amount and focus of intervention of the user in the ontology learning process.

Abraxas, by default, uses the web as a corpus. The notion of ‘information focus’ guides the gathering of new documents from the web in order to augment the original core or seed corpus[2] (if given at all). Information focus may be seen as a view over the three resources which determines exactly what information should be used to generate search engine queries to the underlying search engine. For instance, the information focus may specify that only triples with confidence above a certain threshold should be used in query generation.

The following user profiles are presently defined: ‘fully automated’, ‘IE expert’, ‘knowledge engineer’, ‘corpus linguist’ and ‘fully manual’. The ‘fully automated’ setting requires no user input, apart from pointing to a script that bootstraps the system by specifying one or more of seed triples/patterns/documents. The ‘IE expert’, ‘knowledge engineer’ and ‘corpus linguist’ settings require user intervention to confirm a proposed value of the likelihood of a pattern, triple or document, respectively. Finally, the ‘fully manual’ setting requires the user to decide about the likelihood of all resources.

The algorithm starts by initializing the above data structures - the profile is set to ‘fully automated’ by default and corpus, ontology and wrapper start empty. Abraxas is implemented as an event-driven system. The system starts

[2] A core corpus, in our terminology, is a given set of documents of significant size, while a seed corpus merely acts as guide to the domain. There is no principled difference.

in a state of equilibrium. When ‘disturbed’ by some new event such as adding a new document to the corpus or a new triple to the ontology, it triggers a new learning cycle where it will try and return to a new state of equilibrium, filling in the knowledge gap between corpus and ontology using existing patterns and/or inducing new ones. Events are triggered by external actions specified by the user either in batch mode or interactively at any time, or by internal scheduling of new actions by the learning process. A scheduler decides on the best action to take whenever the system has not reached a state of equilibrium and there is no input of an external action. This is dependent largely on the current state of the KG and EKG measures, which are continuously updated following each relevant action. Simple versions of the scheduler work according to the chosen user profile - for instance, in the ‘IE expert’ setting, the scheduler will give priority to adding new patterns into the system before starting to add new ontology triples or documents. In the future, we will look at implementing a scheduler that examines how the overall likelihood of the data changes with performing a given action, and chooses the best action accordingly. In the following, we describe the actions triggered by each possible event.

Adding a document to the corpus. This triggers extracting some features from the document, such as part-of-speech tags, orthography, gazetteer and named entity tags. The features are represented in a graphical model of the document. The likelihood value of the newly added document is either specified or calculated with respect to the existing resources. Depending on the profile chosen, the user may be requested to confirm such a value. An action is internally scheduled to update the wrapper (induce new patterns) given the now augmented corpus.

Adding a pattern to the wrapper. This triggers representing the pattern as a graph walk over the graph representation of the documents. A graph walk is a set of operations over a set of initial input nodes in a graph, yielding a set of output nodes. Based on the canonical edge traversal operation, operations include node set union and intersection, node substitution and (sub)walk repetition[3]. The likelihood value of the newly added pattern is either specified or calculated with respect to the existing resources, and depending on the profile chosen, the user may be requested to confirm such a value. An action is internally scheduled to update the ontology (place triples into ontology) given the now augmented wrapper.

Adding a triple to the ontology. This triggers the insertion of the triple in the ontology. This results in the ontology structure being re-arranged so that it remains valid, e.g. a concept cannot be both sibling and child of another. The likelihood value of the newly added triple is either specified or calculated with respect to the existing resources, and depending on the profile chosen, the user may be requested to confirm such a value. An action is internally scheduled to update the corpus (add a new document from the web) given the now augmented ontology and in light of the current state of the KG/EKG measures.

Adding new corpus/wrapper/merging ontology. These simply trigger multiple add document/pattern/triple events.

Updating the wrapper. Patterns are induced using an algorithm based on (Ciravegna, 2001), adapted to work over the graph representation of the documents. The pattern induction algorithm starts by spotting all co-occurences of the subject-object pairs found in the triples in the ontology. Each co-occurence instantiates an initial graph walk, which is generalized by dropping edge traversals from the walk and introducing null-cost traversals for skips, allowing for, e.g., skipping adjectives. Candidate new patterns are ranked and only the topmost pattern is selected for augmenting the wrapper. An internal action is scheduled to update the likelihood of all resources.

Updating the ontology Acquiring new ontology triples is done through application of the wrapper to the corpus. This step also requires employing syntactic and semantic similarity measures to cluster the subject/objects of the triples occurring in the documents. Candidate new triples are ranked and only the topmost triple is selected to be merged into the ontology. An internal action is scheduled to update the likelihood of all resources.

Updating the corpus. Adding new documents to the corpus is done by simply requesting the top ranked document from a candidate document queue gathered by an independent web harvester process. The web harvester works as follows. The candidate documents in the queue, limited to a user-defined size, are ranked according to their likelihood of belonging to the domain (which is calculated the same way as for documents in the corpus). The process is one of ever-refining the quality of the queue by looping over each candidate document from the web, calculating its likelihood and deciding whether to keep it or discarding it. Whenever the ontology changes, the likelihood of the documents in the queue is re-calculated and the generated search engine queries potentially change as well. Query generation is achieved by instantiating highly likely patterns in the wrapper with highly likely ontology triples. Presently we use Google API to perform the search.

Updating the likelihood. The likelihood of one resource type is derived from the likelihood of a set of resources of the other two types. For instance, the likelihood of an individual pattern is a function of the likelihood of the triples that pattern is able to spot in the corpus, and of likelihood of the documents where it spots those triples. The likelihood formula for patterns was designed with the following properties in mind. Firstly, to combine into one single formula the effect of both triples and documents; secondly, patterns that cover more triples/documents should be assigned higher likelihood; finally, patterns that cover highly likely triples/documents should be assigned a higher likelihood.

Let O be the set of co-occurrences of subject/object pair of triples t_o in documents d_o. Let O_p be the set of co-occurences restricted to those co-occurrences matched by the pattern p in question.

$$t_p = \sum_{o \in O} conf(t_o), t_r = \sum_{o \in O_p} conf(t_o) \qquad (3)$$

[3] For more details on the graph representation of documents and graph walks, cf. http://wit.shef.ac.uk/runestone

$$t_n = \sum_{o \in O} 1 - conf(t_o), t_w = \sum_{o \in O_p} (1 - conf(t_o)) \quad (4)$$

Similar functions can be defined for d_r, d_w, d_p and d_n.

$$\begin{aligned} r &= t_r + \frac{(t_n - t_w)}{((t_p + t_n) - (t_r + t_w))} + \\ &\quad d_r + \frac{(d_n - d_w)}{((d_p + d_n) - (d_r + d_w))} \end{aligned} \quad (5)$$

$$\begin{aligned} w &= t_w + \frac{(t_p - t_r)}{((t_p + t_n) - (t_r + t_w))} + \\ &\quad d_w + \frac{(d_p - d_r)}{((d_p + d_n) - (d_r + d_w))} \end{aligned} \quad (6)$$

$$lh(p) = \frac{r}{(r + w)} \quad (7)$$

The likelihood of triples and documents is determined analogously (cf. Eq. 7). It suffices to replace O_p by O_t, the set of co-occurrences restricted to a given triple, or O_d, the the set co-occurrences restricted to a given document, respectively.

If only seed documents are given as input to the algorithm (no seed patterns/triples), term recognition techniques are employed in order to determine the most relevant terms and used as seed triples. The algorithm terminates upon user request or whenever a certain condition on the resources is met. Presently, conditions supported are thresholds on size of corpus, size of ontology, likelihood of corpus and likelihood of ontology. Note that no data is ever removed from the system from start to algorithm termination - at most, data may be deemed irrelevant and used as a counter-example.

5. Related Work

The literature on ontology learning is extensive. Here we concentrate on the most important work that has influenced our thinking. The original inspiration for using lexico-syntactic patterns is Hearst (1992) especially as developed by Morin (1999). A number of authors have worked on ways to build ontologies accessing resources beyond the original corpus. Our own work (Brewster et al., 2003) left the range of possible sources open, while (Cimiano et al., 2005) specifically experiment with using data from WordNet, the Web (in general) and the counts provided by Google. However, their approach is focused on replicating an existing ontology and has a linear or pipeline architecture. Etzioni et al. (2005) have created the KnowItAll system which uses exclusively the Web to collect factual information. This system is not designed to construct ontologies but rather isolated facts, and in one version to learn class names (Popescu et al., 2004). They use Hearst type patterns to extract facts and use Pointwise Mutual Information to assess the probability of the extracted facts being correct. The system uses pattern learning to extend the set of predefined lexico-syntactic patterns in an iterative manner. With pattern learning, KnowItAll becomes a bootstrapped learning system, where rules are used to learned new seeds, which in turn are used to learn new rules.

Yangarber *et al.* (Yangarber et al., 2000) originally proposed the use of seed patterns to iteratively expand the set of patterns used for information extraction. However, they did not conceive of automatically expanding the corpus.

The Armadillo system developed at Sheffield (Ciravegna et al., 2004) is again a system which does not learn or construct ontologies but makes extensive use of multiple information sources to build up a database of facts. Armadillo makes extensive use of the redundancy of the Web in order to build up a confidence or likelihood value in each fact it identifies. It is designed to automatically identify potentially relevant 'oracles' for the domain for which it has been tailored and over time expand the set of resources it uses.

6. Conclusion

We have presented a case for a radically different approach to the process of ontology learning in the light of both our view on the nature of knowledge and the relationship between texts and the ontology that they provide evidence for. We presented our approach to measuring the *knowledge gap* between a given ontology and a given set of texts and how this can be extended to the notion of an explicit knowledge gap. Then in the context of the overall algorithm which Abraxas uses we have shown how to conceptualise a tripartite approach to ontology learning focusing on the set of texts, the set of learning patterns and the set of ontological triple.

Future work will look at how to refine the measures which evaluate the equilibrium between the difference resources, and to develop means to evaluate the system as a whole.

7. Acknowledgement

This worked has been undertaken in the context of the Abraxas project (www.nlp.dcs.shef.ac.uk/abraxas) funded by the UK Engineering and Physical Sciences Research Council under grant number GR/T22902/01.

8. References

Khurshid Ahmad. 1995. Pragmatics of specialist terms: The acquisition and representation of terminology. In *Proceedings of the Third International EAMT Workshop on Machine Translation and the Lexicon*, pages 51–76, London, UK. Springer-Verlag.

Christopher Brewster and Kieron O'Hara. 2006. Knowledge representation with ontologies: Present challenges - future possibilities. *International Journal of Human-Computer Studies*. (Introduction to Special Issue).

Christopher Brewster and Yorick Wilks. 2004. Onologies, taxonomies, thesauri: Learning from texts. In Marilyn Deegan, editor, *The Keyword Project: Unlocking Content through Computational Linguistics*. Centre for Computing in the Humanities, Kings College London. Proceedings of the The Use of Computational Linguistics in the Extraction of Keyword Information from Digital Library Content: Workshop 5-6 February.

Christopher Brewster, Fabio Ciravegna, and Yorick Wilks. 2003. Background and foreground knowledge in dynamic ontology construction. In *Proceedings of the Semantic Web Workshop, Toronto, August 2003*. SIGIR.

Christopher Brewster, Harith Alani, Srinandan Dasmahapatra, and Yorick Wilks. 2004. Data driven ontology evaluation. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC-04)*, Lisbon, Portugal.

Philipp Cimiano, Aleksander Pivk, Lars Schmidt-Thieme, and Steffen Staab. 2005. Learning taxonomic relations from heterogeneous sources of evidence. In *Ontology Learning from Text: Methods, Evaluation and Applications*, Frontiers in Artificial Intelligence. IOS Press.

Fabio Ciravegna, Sam Chapman, Alexiei Dingli, and Yorick Wilks. 2004. Learning to harvest information for the semantic web. In Christoph Bussler, John Davies, Dieter Fensel, and Rudi Studer, editors, *ESWS*, volume 3053 of *Lecture Notes in Computer Science*, pages 312–326. Springer.

Fabio Ciravegna. 2001. $(\mathrm{LP})^2$, an adaptive algorithm for information extraction from web-related texts. In *Proceedings of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining held in conjunction with the 17th International Joint Conference on Artificial Intelligence*.

Oren Etzioni, Michael J. Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artif. Intell.*, 165(1):91–134.

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

Diana Maynard and Sophia Ananiadou. 2000. TRUCKS: a model for automatic term recognition. *Journal of Natural Language Processing*, December.

Emmanuel Morin. 1999. Using lexico-syntactic patterns to extract semantic relations between terms from technical corpus. In *5th International Congress on Terminology and Knowledge Engineering Innsbruck (Austria) 23-27 August 1999*, pages 268–278.

Ana-Maria Popescu, Alexander Yates, and Oren Etzioni. 2004. 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.

Willard V. Quine. 1951. Two dogmas of empiricism. *The Philosophical Review*, 60:20–43. Reprinted in W.V.O. Quine, From a Logical Point of View (Harvard University Press, 1953; second, revised, edition 1961.

Marta Sabou. 2004. From software APIs to web service ontologies: A semi-automatic extraction method. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, *International Semantic Web Conference*, volume 3298 of *Lecture Notes in Computer Science*, pages 410–424. Springer.

Shih-Hung Wu and Wen-Lian Hsu. 2002. Soat: A semi-automatic domain ontology acquisition tool from chinese corpus. In *COLING 2002, 19th International Conference on Computational Linguistics, Proceedings of the Conference, 2 Volumes, 24 August - 1September, 200, Academia Sinica, ACLCLP, and National Tsing Hua University, Taiwan*. Morgan Kaufmann.

Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. 2000. 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.