Acquisition of Hypernyms and Hyponyms from the WWW

Ratanachai Sombatsrisomboon [*1] Yutaka Matsuo [*2] Mitsuru Ishizuka [*1]

[*1] Department of Information and Communication Engineering
School of Information Science and Technology
University of Tokyo

[*2] National Institute of Advanced Industrial Science
and Technology (AIST)

ratchai@miv.t.u-tokyo.ac.jp

Abstract. Recently research in automatic ontology construction has become a hot topic, because of the vision that ontology will be the core component to realize the semantic web. This paper presents a method to automatically construct ontology by mining the web. We introduce an algorithm to automatically acquire hypernyms and hyponyms for any given lexical term using search engine and natural language processing techniques. First, query phrase is constructed using the frame *"X is a/an Y"*. Then corpora sentences is obtained from the result from search engine. Natural language processing techniques is then employed to discover hypernym/hyponym lexical terms. The methodologies proposed here in this paper can be used to automatically augment natural language ontology, such as WordNet, domain specific knowledge.

1. Introduction

Recently research in ontology has been given a lot of attention, because of the vision that ontology will be the core component to realize the next generation of the web, Semantic Web [1]. For example, natural language ontology (NLO), like WordNet [2], will be used to as background knowledge for a machine; domain ontology will be needed as specific knowledge about a domain when particular domain comes into discussion, and so on.

However, the task of ontology engineering has been very troublesome and time-consuming as it needs domain expert to manually define the domain's conceptualization, left alone maintaining and updating ontologies.

This paper presents an approach to automatically acquire hypernyms and hyponyms for any given lexical term. The methodologies proposed can be used to augment domain specific natural language ontology automatically or as a tool to help constructing the ontology manually.

2. Related Work

There has been studies on extracting lexical relation from natural language text. The work by Hearst [3] introduces an idea that hyponym relation can be extracted from free text by using predefined lexico-syntactic patterns, such as "NP_0 *such as* {NP_1, NP_2 … , *(and | or)*} NP_n" or "*NP* {,} *including* {*NP* ,}* {*or | and*} *NP*", and so on. For example, in the former pattern, the relations *hyponym*(NP_i, NP_0)*; for all i from 1 to n* can be inferred. With these predefined patterns, hyponym relations can be obtained by running an acquisition algorithm using pattern matching through text corpus.

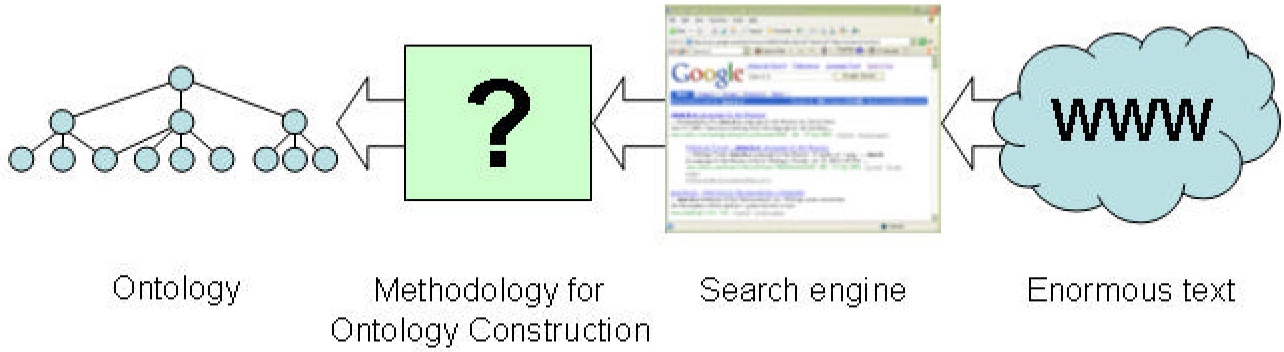

Fig. 1. New approach to Ontology Construction from the WWW using search engine

Another lexical relation acquisition using pattern matching proposed by Sundblad [4]. The approach in [4] extracts hyponym and meronym relation from question corpora. For example, *'Maebashi'* can be inferred as a location from a question like *"Where is Maebashi?"* However, question corpora are very limited in amount since question is less frequently used in normal text. Moreover, relations that can be acquired form question are very limited.

Both methodologies in both [3] and [4], acquires whatever relation found in the text corpora. Because matching pattern on large text corpora consumes a lot of machine processing power and time, therefore relation of specific interest of specific concept cannot be specifically inquired. The methodology to solve this problem is described in the next section.

3. Our Approach

Huge amount of information is currently available on the WWW and increasing at a very high rate. At the time of writing, there are more than 3 billion web pages on the web with more than one million web pages are added daily. Web users can get to these pages by querying specific term(s) to search engine. With index structure of currently available search engine, it is possible to form a more specific a query with phrasal expression, boolean operation, and etc. The result returned by search engine contains a URL and a sentence that the query appears with their context, which is called snippet. All of above inspired us the new approach in acquiring lexical relations, which will be introduced next.

The web is huge, and therefore we want to use it as our corpus to discover lexical relations, but neither obtaining all the text from the web, nor processing it is possible. Therefore we construct a small corpus according to query on-fly from putting sentences together. Corpora sentences are extracted from the query result's snippets, which means there is no interaction with web page's host, only interaction with search engine index is needed. However, to obtain corpora sentences that can be used to extracted lexical relation, we need to formulate a query according to the pattern that we will use to extract relation to a search engine. For example, if we will use "*X is a/an Y*" pattern to extract hyponym relation between *X* and *Y*, then we build up a query as the phrase "*X is a/an*" or "*is a/an Y*".

After the corpora sentences have been obtained, pattern matching and natural language processing technique can be applied to discover lexical relation, and then statistical technique will be employed to guarantee the accuracy of the result. Finally, domain ontology can be constructed. (see figure 1).

4. Acquiring Hypernyms

From the definition of hyponymy relation written in [2] – a concept represented by the synset {*x*, *x*', …} is said to be a hyponym of the concept represented by the synset {*y*, *y*', …} if native speaker of English accept sentences constructed from such frames as "*An x is a (kind of) y*", we query to search engine using the query phrase "*X is a/an*" to acquire hypernyms for *X*, For example, we query to search engine with the phrase "*scripting language is a/an*" to find hypernyms for the "*scripting language*". For each result returned from search engine, the sentence that contains the query phrase is then extracted from snippet. Example of these sentences can be seen in figure 2.

From those sentences we then filter out undesirable sentences, e.g. sentences marked by * in figure 2, and extract lexical items that are conceivable as query term's hypernyms (The terms that appear in bold face in the figure). In filtering out the undesirable sentences, we remove the sentences that is the query phrase is not the start of the sentence or not preceded by conjunctions such as 'that', 'because', 'since', 'while', 'although', 'though', 'even if', and etc. For extracting the hypernyms term, we first tag all terms with POS and capture the first noun (or compound noun) in the noun phrase ignoring its descriptive adjective.

Extracted lexical items that represents the same concept are then grouped together according to synonymy (synsets) defined in WordNet. Finally, the concepts that occur more frequently proportion to the others are then suggested as hypernyms for the query. As an example, from figure 2., we can infer relation *hypernym("scripting language", "programming language")*, since 'programming language' is the lexical concept that appear most frequently.

A *scripting language is a* lightweight **programming language.**

Scripting language, is an interpreted **programming language** that …

I think that a *scripting language is a* very limited, **high-level language** .

Since a *scripting language is a* full function **programming language**

* The BASIC *scripting language is a* dream to work with.

* The *scripting language is a* bit tough for me, esp.…

Fig. 2. Example of sentences extracted from results obtained from a search engine for query phrase "scripting language is a/an".

5. Acquiring Hyponyms

The algorithm to acquire hyponyms is similar to acquiring hypernyms in section 4. It begins with formulate a query phrase, query to search engine, and then obtain the results which will be used as a corpus to extract hyponyms.

Same as the in acquiring hypernyms, we exploit the frame of "*An x is a (kind of) y*" to discover hyponyms. To discover hyponyms for lexical concept *Y*, we first construct query phrase, "*is a/an Y*", and query to a search engine. As an example, the sentences extracted from returned result for the query "*is a scripting language*" are shown in figure 3.

XEXPR *is a scripting language* that uses XML as…

I know that **python** *is a scripting language,* but I'm not sure…

JavaScript *is a scripting language* used in Web pages, similar to…

Tell your friend that **C** *is a scripting language* too.

* What is *a scripting language*?

* A language is decided upon whether it *is a scripting language* by…

Fig. 3. Example of sentences extracted from results obtained from a search engine for query phrase "is a scripting language"

After we have extracted the sentence out of each snippet, and filter out the sentence like the one marked by * in figure 3., lexical items that comes right before the query phrase *"is a/an Y"* are spotted as candidate hyponyms of *Y*. (Shown as bold face in the figure.)

Subsequently, each candidate with small number of occurrence is then confirmed by acquiring its hypernyms and check if concept *Y* is in its top hypernyms acquired. If it is, then the candidate term is accepted as hyponym. For example, in the figure 3, there is a statement that 'C' is a scripting language, however, as you might know 'C' is actually not a scripting language (or if it is, definitely it is not well recognized as a scripting language and thus should there be no formal semantic relation between the terms). Therefore, 'C' will be rejected in the confirmation process since scripting language will not be in the top hypernyms of 'C'.

Finally, the candidate terms that have large number of occurrence and the candidate terms that pass the confirmation process are suggested as hyponyms for *Y*.

6. Examples

In this section, we report an experiment of our proposed algorithm for acquiring hypernyms and hyponyms. The system is implemented with Perl using Google Web APIs [5] as an interface with index of Google search engine[6]. The number of corpora sentences retrieved from the search engine ranges from zero to more than thousand sentences. We avoid a problem of reliability of information source by limit maximum number of sentences extracted from a particular domain to 2 sentences. The result of experiment for query terms that yield outputs are shown in figure 4 and 5.

In figure 4, given query terms as input, a list of their hypernyms can be derived. There are large amount of information regarding the first three query terms on the web, in which a lot of *"X is a/an NP"* sentence pattern can be extracted (number of corpora sentences extracted is written in brackets next to the query term), and thus yield very accurate results, as the system acquires 'programming language' and 'language' as hypernyms for query terms 'Java', 'Perl', and 'Python' with highest percentage relative to acquired hypernyms (number on the right of acquired hypernyms shows proportion of number of sentence the hypernym appears with number of total corpora sentences expressed as a percentage). For the query terms 'Maebashi' and 'Active Mining', which less information is available on the web (in English text, to be accurate), there is only a small number of corpora sentences can be extracted. Nevertheless, the result hypernyms yielded are still accurate as it can tell that 'Maebashi' is a city, however for 'Active Mining', 'new direction' is the result because 3 out of 5 corpora sentences are "*Active mining is a new direction in data mining/the knowledge discovery process*"

The result of applying hyponyms acquisition algorithm proposed in this paper can be seen in figure 5. The query terms are 'programming language', 'scripting language', and 'search engine'. Each of these lexical concepts represent a very large class with a lot of members, which a number of those members can be discovered as shown in the figure. The precision of the acquired hyponyms is quite satisfying as shown in the acquired result, however we do not show the recall measure here, and thus it will be our future work on result evaluation.

7. Discussion and Conclusions

We have introduced a new approach to automatically construct ontology using search engine, natural language processing techniques. Methodologies for acquiring hypernyms and hyponyms of a query term are described. With these two techniques, taxonomy of domain ontology as shown in figure 6 and alike can be automatically constructed in a very low cost with no domain-dependent knowledge is required. Finally, the domain specific ontology constructed can then be augmented to natural language ontology (NLO), e.g. WordNet.

QUERY TERM	ACQUIRED HYPERNYMS
JAVA (1011)	programming language(0.33), language(0.20), object-oriented language(0.06), interpreted language(0.05), trademark(0.05)
PERL (913)	programming language(0.23), language(0.23), interpreted language(0.15), scripting language(0.11), tool(0.03), acronym(0.03)
PYTHON (777)	programming language(0.37), language(0.20), scripting language(0.14), interpreted language(0.06), object-oriented language(0.03)
SEARCH ENGINE (84)	tool(0.13), index(0.07), site(0.06), database(0.06), program(0.6), searchable database(0.05)
SCRIPTING LANGUAGE (23)	programming language (0.35)
MAEBASHI (11)	City (0.63), international city (0.45)
ACTIVE MINING (5)	new direction (0.60)

Fig. 4. Hypernyms acquired for selected query terms

PROGRAMMING LANGUAGE : ABAP, ADA, APL ,AppleScript, awk, C, CAML, Cyclone, DarkBASIC, Eiffel, Erlang, Esterel, Expect, Forth, FORTRAN, Giotto, Icon, INTERCAL, Java, JavaScript, Kbasic, Liberty BASIC, Linder, Lisp, LOGO, Lua ,ML, Mobile BASIC, Modula-2, Nial, Nickle, Occam ,Pascal, Perl, PHP, Pike, PostScript, Prolog, Python, Quickbasic, Rexx, Smalltalk, SPL, ToonTalk, Turing, VBScript, VHDL, Visual DialogScript, XSLT

SCRIPTING LANGUAGE : AppleScript, AREXX, ASP, AWK, CCSH, CFML, CFSCRIPT, ColdFusion, Compaq Web , anguage, CorbaScript, DelphiWebScript, ECMAScript, Expect, FDL, ferrite, Glish, IDLScript, JavaScript, Jint, Jscript, KiXtart, ksh, Lingo, Lite, Lua, Lucretia, Miva Script, MML, Perl, Pfhortran, PHP, PHP3, Pnuts, Python, REXX, Ruby, RXML, STEP, Tcl, Tcl/Tk, UserTalk, VBScript, WebScript, WinBatch

SEARCH ENGINE: Aeiwi, AFSearch, AlltheWeb, AltaVista, antistudy.com, Ask Jeeves, ASPseek, Biolinks, Convonix, CPAN Search, Dataclarity, DAYPOP, FDSE, Feedster, FileDonkey, Fluffy Search, FreeFind, GamblingSeek, Google, HotBot, Inktomi, kids.net.au, Londinium.com, Mirago, mnoGoSearch, Northern Light, OttawaWEB, Overture, Phantis, PsychCrawler, Scirus, Search Europe, search4science, SearchNZ, Searchopolis.com, SiteSearch, SpeechBot, Teoma, Vivisimo, WebCrawler, WebWombat, XL Search, Yahoo!, Yahooligans!

Fig. 5. Hyponyms acquired for query term programming language, scripting language, and search engine

Moreover, this methodology requires only small amount of time (in the matter of seconds for hypernyms acquisition or minutes for hyponyms including confirmation process) to discover the query's subset/superset concepts, and thus domain specific lexical concept can be learned on-fly (given the term is described somewhere on the web with *"is a/an"* pattern, and has been indexed by the search engine employed in the system).

Although the work reported in this paper uses only the pattern *"NP is a/an NP"*, we can also use a set of lexicon syntactic patterns suggested by Hearst [3] to acquire hyponyms of a query term. For example, we can use a pattern "NP_0 *such as* {NP_1, NP_2 …, (and | or)} NP_n" to formulate the query to search engine as *"Y such as"* and derived corpora sentences to subsequently extract the hyponyms of query term *Y*. However, as a trade-off for the web's growth rate and its size, there is innumerable natural language text on the web that is highly informal, unstructured or unreliable. For that reason, using only pattern matching alone will result in low precision. To solve this problem, we treat terms extracted from pattern matching as candidate terms for hyponym, and then confirm if a candidate term is actually the query term's hyponym by acquiring their hypernyms using method described in section 4, and then check with their acquired hypernyms as suggested in section 5.

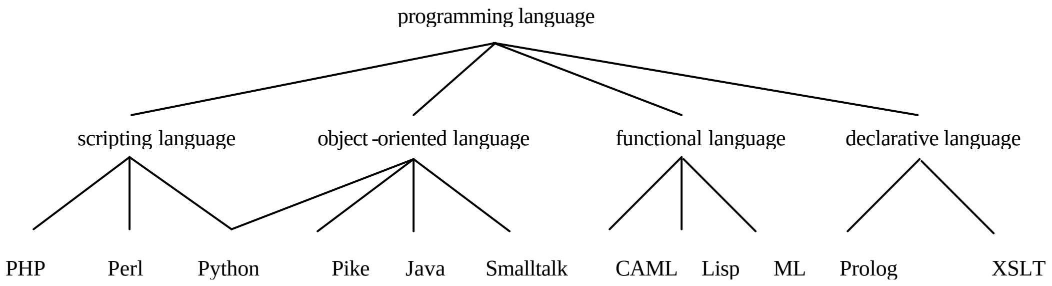

Fig. 6. Example of taxonomy built using proposed methodology

Our method works very well for specific terms. However, it often fail in acquiring hypernyms of general noun, such as 'student', 'animal', and etc. because descriptive sentence with a *"is a/an"* pattern is rarely appear in normal text. Nevertheless, acquiring hypernyms for general terms is hardly of any interest, since general term usually has already been defined in well-established machine understandable dictionary such as WordNet.

8. Future Work

Firstly, we need to formulate an evaluation method based on precision and recall of acquired hypernyms and hyponyms. Secondly, In addition to acquiring lexical term, we also want to acquire its meaning, and thus using corpora sentence's context to disambiguate word sense is our interest for future study. Lastly, apart from hyponymy or ISA relation, there are other semantic relations that we are interested to automate the acquisition process, such as meronymy or HASA relation, and non-taxonomic relation such as *'created by'*, *'produce'*, and etc.

References

1. T. Berners-Lee, J. Hendler, and O. Lassila, The semantic web. In *Scientific American*, May 2001.
2. G. Miller, R. Beckwith, C. Fellbaum, D Gross, and K. Miller. Five papers on wordnet. Technical report, Stanford University, 1993.
3. M. A. Hearst, Automatic Acquisition of Hyponyms from Large Text Corpora, in *Proceedings of the Fourteenth International Conference on Computational Linguistics,* Nantes, France.
4. H. Sundblad, Automatic Acquisition of Hyponyms and Meronyms from Question Corpora, in Proceedings of the Workshop on Natural Language Processing and Machine Learning for Ontology Engineering at ECAI'2002, Lyon, France.
5. Google Web APIs, http://www.google.com/apis/
6. Google, http://www.google.com
7. B. Omelayenko, Learning of ontologies for the Web: the analysis of existent approaches. In *Proceedings of the International Workshop on Web Dynamics*, 2001.
8. E. Agirre, O. Ansa, E. Hovy and D. Martinez, Enriching Very Large Ontologies Using the WWW, in *Proceedings of the Ontology Learning Workshop, ECAI,* Berlin, Germany, 2000.