International Journal of Software Engineering
and Knowledge Engineering
Vol. 24, No. 8 (2014) 1159–1175

DOI: 10.1142/S0218194014400099

World Scientific
www.worldscientific.com

Learning Non-Taxonomic Relations on Demand for Ontology Extension

Yan Xu[*,†,‡], Ge Li[*,§], Lili Mou[*,¶] and Yangyang Lu[*,‖]

[*]*Laboratory of High Confidence Software Technologies (MoE)*
Institute of Software, School of EECS, Peking University
Beijing 100871, P. R. China

[†]*Academy of Mathematics and System Science*
Chinese Academy of Sciences
Beijing 100190, P. R. China
[‡]*xuyan14@pku.edu.cn*
[§]*lige@sei.pku.edu.cn*
[¶]*moull12@sei.pku.edu.cn*
[‖]*luyy11@sei.pku.edu.cn*

Learning non-taxonomic relations becomes an important research topic in ontology extension. Most of the existing learning approaches are mainly based on expert crafted corpora. These approaches are normally domain-specific and the corpora acquisition is laborious and costly. On the other hand, based on the static corpora, it is not able to meet personalized needs of semantic relations discovery for various taxonomies. In this paper, we propose a novel approach for learning non-taxonomic relations on demand. For any supplied taxonomy, it can focus on the segment of the taxonomy and collect information dynamically about the taxonomic concepts by using Wikipedia as a learning source. Based on the newly generated corpus, non-taxonomic relations are acquired through three steps: a) semantic relatedness detection; b) relations extraction between concepts; and c) relations generalization within a hierarchy. The proposed approach is evaluated on three different predefined taxonomies and the experimental results show that it is effective in capturing non-taxonomic relations as needed and has good potential for the ontology extension on demand.

Keywords: Learning on demand; ontology extension; non-taxonomic relations; information retrieval; dependency parsing.

1. Introduction

In computer science, an ontology is defined as a "formal, explicit specification of a shared conceptualization" [1]. Comparing with plain text, ontologies organize information in a structured organization and thus facilitate the sharing and reuse of knowledge. Nowadays, ontologies play an important role in many knowledge-intensive areas such as software engineering [2], e-commerce [3], biomedical

informatics [4] and medicine [5]. It is widely accepted that the manual construction of ontologies is a resource-intensive and time-consuming task [6]. Therefore, ontology learning becomes an emerging field to support the automatic engineering of ontologies. Ontology learning consists of three subtasks, i.e. the lexical entries extraction, the taxonomic relations learning and the non-taxonomic relations learning. The lexical entries extraction captures terms that can be used as ontological concepts; the taxonomic relations learning organizes obtained ontological concepts into a taxonomy; and the non-taxonomic relations learning attempts to discover potential arbitrary relations in the learned taxonomy [7]. This paper is about learning the non-taxonomic relations for ontology extension, which is a hot and challenging research topic at present.

In general, non-taxonomic relations model interactions between the ordered pairs of ontological concepts within an ontology. They are represented in the form of verbs or verb phrases [8, 9]. Comparing with taxonomic relations, non-taxonomic relations have less explicit manifestation as well as more diverse expression. For instance, considering "computer algorithm" topic, the relations between two concepts "greedy algorithm" and "optimization problem" can be presented by the following expressions: "greedy algorithm find solution for optimization problem", "greedy algorithm is used in optimization problem" or "greedy algorithm solve optimization problem". Thus, non-taxonomic relations learning needs to resolve the problem caused by implicity and diversity.

The hybrid of statistics-based techniques and linguistics-based techniques is proposed to be a promising way for learning the non-taxonomic relations [6, 10]. As of now, most of the available statistical analysis approaches employ expert crafted corpora, which are often domain-oriented (e.g. the GENIA Corpus built for Gene domain [11], or BioInfer corpus built for the Biomedical domain [12]). However, on the one hand, such manual mode is laborious and costly; on the other hand, for various specialized taxonomies, static corpora are hard to meet their different demands of semantic relations discovery. Specially, automatic taxonomy induction may carve a taxonomy tailored to a certain document collection [13, 14], or may capture taxonomic information according to specific tasks or personalized requirements [15, 16]. Such generated taxonomies may focus on a certain region of a domain, or may span different domains. In these cases, the above mentioned corpora may not flexible enough. Therefore, an effective approach that can learn non-taxonomic relations on demand might be helpful to ontology extension.

In this paper, we present a novel approach for acquiring non-taxonomic relations. For a target taxonomy to be extended, this approach focuses on taxonomic concepts contained therein, and collects information dynamically around these taxonomic concepts by using Wikipedia as a learning source. In addition, based on the generated corpus, it views learning non-taxonomic relations as a process consisting of three clearly defined steps: a) semantic relatedness detection; b) relations extraction between concepts; c) relations generalization within a

hierarchy. In these stages, techniques in information retrieval and natural language processing are integrated to handle with the implicity and diversity of non-taxonomic relations.

To conclude, the contribution of this paper is two-fold. First, defines a learning framework for learning non-taxonomic relations on demand, i.e. for any supplied taxonomy, tailor-makes a corresponding corpus and explores its non-taxonomic relations automatically. Second, from a systematic point of view, designs a three-step pipeline for learning. The proposed approach is evaluated on three different predefined taxonomies and the experimental results shows that our approach is effective in capturing non-taxonomic relations as needed and has good potential for ontology extension.

The rest of this paper is organized as follows. Section 2 discusses related work on ontology learning and relation extraction. Section 3 describes the proposed approach for learning non-taxonomic relations in detail. Several experiments and in-depth analysis are presented in Sec. 4. Finally, Sec. 5 closes with a conclusion of our research and points out ideas for future work.

2. Related Work

Since 2000, ontology learning has attracted the attentions of the researchers. Many of the work focus on learning ontologies from static corpora constructed by domain experts manually. The association rule mining and the dependency parsing are used for learning non-taxonomic relations.

Madche and Staab [17] apply a generalized association rule algorithm and represent co-occurrences of words within a sentence as transactions. It can not only detect relations between concepts, but also determine the appropriate level of abstraction at which to define relations. Ciaramita *et al.* [8] treat syntactic dependencies as potential relations. The dependency paths are scored with statistical measures of correlation by using χ^2-test. At the same time, the abstraction of the relations can be generalized to certain level by using selection restriction algorithm. Similar approaches are used by Schutz and Buitellaar for extending the SportEventOntology [9]. The approaches are based on manually built corpus and require labeled data.

Recently, people find that the information on the Web can be used for ontology learning. Banko *et al.* [18] developed TextRunner to extract information across different domains from the Web. It uses a conditional random field-based model to label the constituents in the input strings whether they are entities or not, and then to capture the relationships between entities to form triples. The extraction process does not require any human input. In addition, the extracted entities and relationships using TextRunner can be used to bootstrap the construction of ontologies. This approach focuses on the general issues in information extraction

field but cannot be applied to conduct the non-taxonomic relations learning directly. In the same domain of open information extraction, Fader *et al.* [19] introduce two simple syntactic and lexical constraints on binary relations for identifying relations further.

Sánchez and Moreno [20] proposed methods for discovering non-taxonomic relations by using search engines. They developed a technique for learning domain patterns using domain-relevant verb phrases extracted from web pages provided by search engines. These domain patterns are then used to extract and label the non-taxonomic relations using linguistic and statistical analysis. Wong *et al.* [21] proposed a hybrid approach based on techniques of lexical simplification, word disambiguation and association inference for acquiring coarse-grained relations between potentially ambiguous and composite terms by using Wikipedia and search engines' page count.

In short, the applications of search engines in [20] and [21] are based on the assumption that the Web approximates the real distribution of the information in human kind, thus the hit count of a search engine can be used for probability estimation. However, such kind of method relies on a specific search engine to a large extent. On the contrary, our approach downloads Wikipedia pages directly and analyzes page information in depth, which can avoid the bias brought by search engines.

The semi-structured data in Wikipedia's category system also capture the attention of researchers in the context of ontology learning. Liu *et al.* [22] proposed an approach named Catriple for automatically extracting triples in Wikipedia's super-sub category pairs. They developed a prototype which has extracted a large number of triples (1.27 M) with high confidence (96%). Confined by the inherent nature of Wikipedia's category structure, the extracted relations are mainly about property and its value (e.g. "Category: Songs by artist"-"Category: The Beatles songs", with "artist" as property and "The Beatles" as value). Recent work related to non-taxonomic relations learning includes Mohamed *et al.* [23] and Serra *et al.* [24]. Mohamed *et al.* [23] propose an approach for automatically discovering relevant relations, given a large text corpus plus an initial ontology defining hundreds of noun categories (e.g. Athlete, Musician and Instrument). Serra *et al.* [24] describe three representative techniques for non-taxonomic relations learning and discuss their advantages and limitations.

3. Non-taxonomic Relations Learning

3.1. *Framework of the approach*

This subsection describes the overview of the learning approach for acquiring non-taxonomic relations. This approach uses a predefined taxonomy as input, and produces a set of ontological triples as output. Each ontological triple contains an

ordered pair of ontological concepts and a semantic relation between them. Formally, we can give the representation structures as follows:

- The input taxonomy is a two-tuples $T := \langle C, H \rangle$, where C is a set of concepts, i.e. $C := \{c_1, c_2, \ldots, c_n\}$, H is a set of subclass relations between C, i.e. $H := C \times C$;
- The output is a set $R := \{\langle c_i, c_j, r_{ij} \rangle | c_i \in C, c_j \in C, r_{ij} \in L\}$, where L is a set of relation names.

The framework of the learning approach is sketched in Fig. 1. The upper part above the dotted line depicts the process of constructing corpus on demand. It takes a predefined taxonomy as input, and collects a corresponding corpus from Wikipedia automatically. The lower part below the dotted line depicts the process of learning non-taxonomic relations. It consists of three learning steps, shown in the big rectangular box. All the learning steps are based on the tailor-made corpus generated by the process in the upper part.

The following sections will detail each component in this approach. In Sec. 3.2, we explain the motivation of treating Wikipedia as an information source and depict its internal search mechanisms for our learning purpose. Then, in Sec. 3.3, we describe

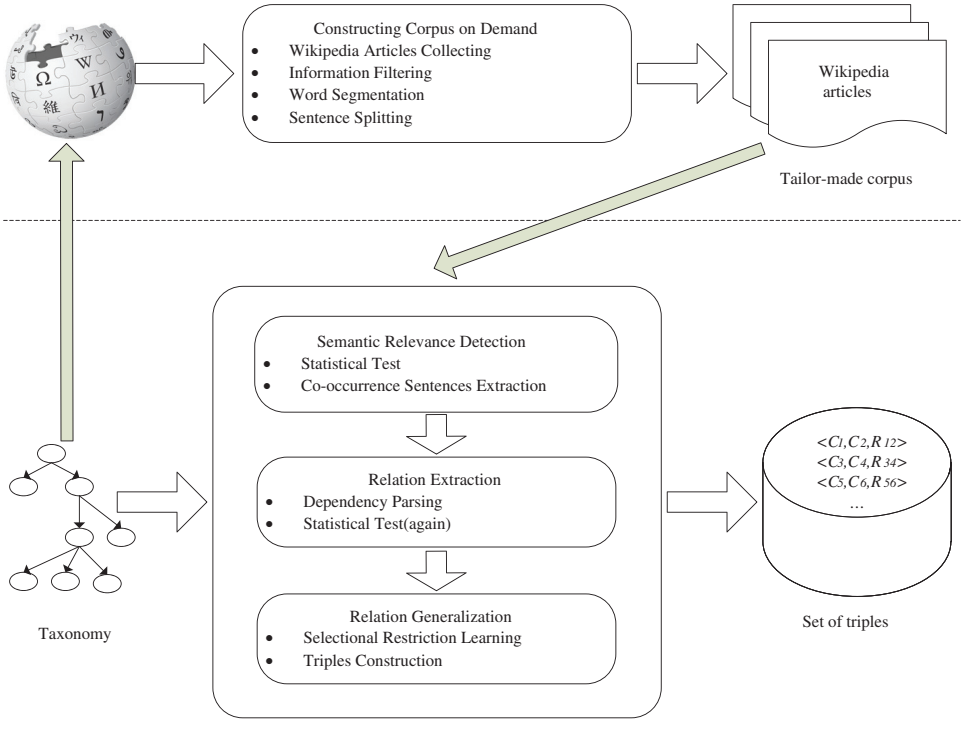

Fig. 1. Framework of the non-taxonomic relations learning approach.

the learning process detailedly. Section 3.3.1 describes how to construct corpus on demand. Section 3.3.2 describes the way for detecting the semantically related concept pairs within the taxonomy by using a statistical test method. Section 3.3.3 applies the dependency parsing to extract relations holding between these ordered ontological concept pairs and uses the statistical test again to choose the preferred relations. Section 3.3.4 conducts the selectional restriction learning for relation generalization. The finally constructed triple collection is the target output of our learning approach.

3.2. *Wikipedia as information source*

In this section, we explain our motivation of treating Wikipedia as an information source. Many ontology learning approaches assume there exist a static domain corpus, which is constructed by domain experts [8, 9]. However, collecting and maintaining such a corpus might not be a trivial project, as information selection and screening is very time-consuming. In addition, the human knowledge demands continuous updating, especially on the open Web platform. Therefore, it is gradually becoming apparent that static and the expert crafted resources may no longer be adequate [6].

Nowadays, the collective intelligence of Web, including WordNet, search engines, Wikipedia and so on, paves the way for such bottleneck. WordNet is an expert-crafted online thesaurus providing semantic knowledge between terms, which is often used as a gold standard for evaluation. It ensures high quality but may be lack of the coverage of up-to-date technical terms. In contrast, search engines provide access for massive information on the Web. The huge size and redundancy of Web data makes it very suitable for simulating the real distribution of the information in humankind. For example, like computing global scale statistics about some targeted information distribution. However, some characteristics like being lack of structure and rich of noise prevent it from further semantic analysis. Therefore, the Web-statistics-based technique is more appropriate for the early stage of ontology learning, e.g. the lexical entry extraction and the taxonomic relations learning [10].

Comparing with WordNet and search engines, Wikipedia makes a trade-off between content quality and information coverage. As an online cyclopedia, Wikipedia provides up-to-date information across vast domains; moreover, the collaborative generation of Wikipedia content makes it reach a community consensus, which aligns well with the principle of ontologies. Thus, we select Wikipedia as a general information source for constructing learning corpora. Specially, we collect the corpora information by using its internal search engine.

Wikipedia Internal Search Engine

According to the internal search mechanism of Wikipedia[a], the retrieval results are a list of pages related to the query keyword. In addition, the relatedness between pages

[a] http://en.wikipedia.org/wiki/Help:Searching

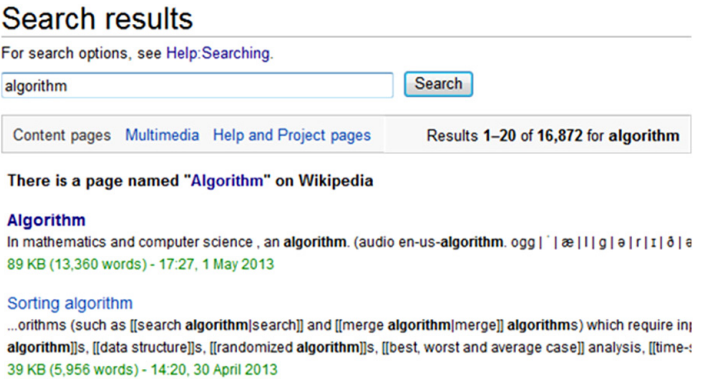

Fig. 2. Retrieval result page of "algorithm" in Wikipedia's internal search engine.

with the query keyword is degrading as the ranking is decreasing. For instance, Fig. 2 illustrates the retrieval results of the query keyword "algorithm".

This internal search mechanism is used in the proposed learning approach for constructing tailor-made corpora by integrating information filtering techniques.

3.3. *Learning process*

3.3.1. *Constructing corpus on demand*

In this subsection, we describe the main steps for tailor making a corpus for a predefined taxonomy. Traditional approaches of constructing domain corpora manually don't view the problem on a conceptual level. That may cause data sparsity for some taxonomic concepts and influence the learning outcome finally. To avoid such problems, we collect a certain number of Wikipedia articles for every taxonomic concept. Furthermore, we conduct a kind of information filtering to filter out articles irrelevant to the given taxonomy.

Firstly, we take all taxonomic concepts as query keywords to search on the internal search engine of Wikipedia, and then crawl a certain number of top pages for every query keyword to form a page collection. It's worth note that when collecting pages about different keywords, we might encounter duplicate pages. Thus duplicate URL detection should be done for selecting different pages altogether.

Secondly, we expect that the document content in the target corpus should all pertain to taxonomy of interest. But the internal search mechanism may retrieve some pages that are weakly related to or not related to target taxonomy. Therefore, information filtering should be conducted to eliminate these pages. For that purpose, we weight information of each Wikipedia retrieval page p_j in accordance with the frequency of taxonomic concept c_i by using the following formula.

$$InforWeight(p_j) = \sum_{i=1}^{m} tf_{ij} \times \log \frac{N}{n_i} \tag{1}$$

where N is the number of pages in page collection and $1 \leq j \leq N$, tf_{ij} is the frequency of c_i occurring in page p_j, n_i is the number of pages containing c_i, m is the number of concepts. According to Eq. (1), the information weight of a page p_j is normalized by the sum of $tf\text{-}idf$ value of concepts in the predefined taxonomy. Information filtering is conducted by retaining those pages p_j whose $InforWeight(p_j) > \theta$, where θ is an empirical threshold. For example, given an taxonomic concept "algorithm" as the query word, the Wikipedia search engine would return a set of related terms, like "parallel algorithm", "simplex algorithm", "proprietary software" and so on. According to the Eq. (1), "proprietary software" might be filtered out for its low information weight.

Finally, by HTML tags removing (by using HTML parser[b]), word segmentation and sentence splitting (by using Stanford parser[c]), we obtain a text collection for one concept $c_i \in C$. All the text collections for the set C compose the tailor-made corpus for the given taxonomy.

3.3.2. *Semantic relevance detection*

As described in Sec. 1, before figuring out the concrete relations between the ordered concept pairs, it is necessary to detect the semantic relevance of the concept pairs. Apparently, not all the concept pairs within the predefined taxonomy have semantic association. For every concept in the concept set, we detect its semantic relevance with every other concept.

Here, the χ^2-test [25] can be used to compute a relevance ranking. For a concept pair, we count their co-occurrence information and represent it with contingency tables, and then calculate the value of χ^2. The simplified formula is given below.

$$\chi^2 = \frac{N(O_{11}O_{22} - O_{12}O_{21})^2}{(O_{11} + O_{12})(O_{11} + O_{21})(O_{12} + O_{22})(O_{21} + O_{22})} \tag{2}$$

where indices refer to the column and row of the table, O is the observed frequency and N is the total number of sentences within the corpus.

For every concept, we apply Eq. (2) on the generated corpus to obtain a list of χ^2 value ordered by relevance in terms of this ranking. By setting a threshold of semantic relevance, the concepts weakly relating to the given concept are eliminated. At the mean time, the sentences including highly relevant concept pairs are retained for further analysis.

3.3.3. *Relation extraction*

After rejecting those concept pairs from the χ^2-sorted list whose score cannot reflect high semantic relevance, we need to figure out the relations between remain concept pairs. As it is described in introduction, verbs or verb phrases are often indicators

[b] http://htmlparser.sourceforge.net/
[c] http://nlp.stanford.edu/software/lex-parser.shtml

expressing relations between concept pairs. From a syntactic parsing point of view, we just expect to find such verb structure, i.e. predicate-argument structure, while dependency parsing provides a sound solution in this aspect.

We choose Stanford parser to check dependency structure of the remaining sentences with concept pairs of high relevance. The parsing result of a sentence by Stanford parser is a list of dependencies. The dependencies are all binary relations, representing by a grammatical relation holding between a governor and a dependent. The list of dependencies for a sentence maps straightforwardly onto a directed graph representation. Words in the sentence are nodes in the graph and dependencies between words are edges with edge labels named by grammatical relations. A word in the sentence may have several modifiers, but each word may modify at most one word (we do not consider the cyclic structure as described in [26]). Figure 3 provides an example of the sentence "Greedy algorithms find the overall, or globally, optimal solution for some optimization problems".

Modern theory of syntax suggests above all that verbs are predicates and the noun phrases that they appear with are their arguments; then, other function words (e.g. auxiliary verbs, certain prepositions and phrasal particles) are viewed as part of the predicates [27]. While in our learning approach, what we want it is to find the predicates according to the given arguments. Intuitively, it is corresponding to look for a connected path between two vertices of the argument in the directed graph.

Among all possible paths, we need to make some restrictions to select the expected predicate-argument structure. Clearly, the pivotal element connecting two arguments must be a root verb of the clause; furthermore, we demand the grammar relations between root verb and argument should be either agentive or objective. This restriction is not limited to a specific dependency parser, while in the case of Stanford parser, agentive grammar relations include ***nsubj*** (normal subject), ***nsubjpass*** (passive nominal subject); objective grammar relations include ***dobj*** (direct object), ***iobj*** (indirect object), ***pobj*** (object of a preposition). Given arg (argument), verb and ***gr*** (grammar relation), we define the expression arg $\leftarrow$ ***gr*** $\leftarrow$ verb meaning that verb governs arg by ***gr***. Five instances of ***gr*** (***nsubj***, ***nsubjpass***,

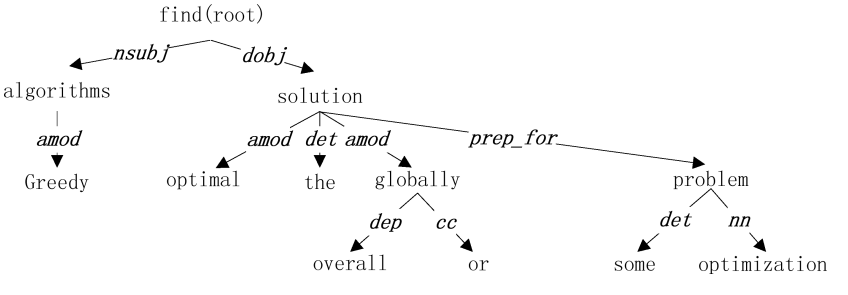

Fig. 3. Dependency parse tree of the sentence "Greedy algorithms find the overall, or globally, optimal solution for some optimization problems".

dobj, ***iobj***, ***pobj***) can compose eight meaningful constrains as follows:

(1) arg1 ← ***nsubj*** ← root verb → ***dobj*** → arg2
(2) arg1 ← ***nsubjpass*** ← root verb → ***dobj*** → arg2
(3) arg1 ← ***nsubj*** ← root verb → ***iobj*** → arg2;
(4) arg1 ← ***nsubjpass*** ← root verb → ***iobj*** → arg2;
(5) arg1 ← ***nsubj*** ← root verb → ***pobj*** → arg2;
(6) arg1 ← ***nsubjpass*** ← root verb → ***pobj*** → arg2;
(7) root verb → ***iobj*** → arg1 → ***dobj*** → arg2;
(8) root verb → ***iobj*** → arg1 → ***pobj*** → arg2;

Once a sentence matches one of these eight constraints, we extract words along the path connecting arg1 and arg2 to compose the target predicate. In the example sentence "Greedy algorithms find the overall, or globally, optimal solution for some optimization problems", for the two argument "Greedy algorithm" and "optimization problem", the target predicate should be "find-solution-for". At last, we apply Eq. (2) of statistical test again to find the preferred predicates strongly associated with the ordered concept pair.

3.3.4. *Relation generalization*

When learning relations, a crucial issue is to find the right level of abstraction for ontological relations with respect to the concept hierarchy. Cimiano *et al.* [28] show a systematic analysis about this problem. They evaluate three different measures from the sub-categorization and selectional restrictions acquisition communities on Genia annotated corpus and Genia ontology. The experimental results show that the conditional probability based measure outperforms the other two measures, namely, pointwise mutual information based measures and χ^2 based measures. Therefore, we adopt conditional probability based measure to find the correct level of generation with respect to a given ontological hierarchy for extracted relations.

The key idea of the conditional probability based measurement is as follows:

- For a certain slot v_s of a verb v and a concept c, calculating the conditional probability that a concept c appears in this slot;
- Conduct such a calculation along the given ontological hierarchy from c upwardly;
- Choose the concept maximizing this conditional probability value.

That can be expressed by:

$$c_{v_s} := \arg\max_{c} P(c \mid v_s) \tag{3}$$

If there are several concepts with the same value, we choose the most specific one, leaving out the concepts which subsume them.

At the end the whole learning process, we obtain a set of triples consisting of relations holding between ordered taxonomic concept pairs as output of this learning approach.

4. Experiments

The evaluation of ontology learning is not a trivial task [29]. So far, many evaluation efforts have been made for on the lexical as well as on the taxonomic level, where gold standard could be found easily. In contrast, on the non-taxonomic level, it is hard to find a uniform standard. Therefore, we have to resort to a manual inspection by experts and select appropriate evaluation metrics.

We experiment on three different taxonomies placed in our semantic repository named *Knowware Library*. Their topics are about "Database", "Operating System" and "Software Engineering", respectively. The "Database" ontology contains 80 concepts, organizing database related concepts into a tree hierarchy. The "Operating System" ontology contains 96 concepts, presenting key concepts in the domain of operating system like memory management, process scheduling and file system. The "Software Engineering" Ontology contains 101 concepts, describing various aspects in software engineering, such as software architecture, software design and software quality. One other thing to note is that existent relations between these taxonomic concepts are not exactly "is-a" subclass relations. There are other semantics like "part-of" and "has-property-of" among these relations. These structural relations provide backbone for learning non-taxonomic relations expressing interactive semantics.

4.1. *Experimental process*

4.1.1. *Corpora building*

We collect Wikipedia articles to build corpora for three chosen taxonomies. We use taxonomic concepts as query words to retrieve related articles in Wikipedia; after downloading a certain number of Wikipedia articles, we do information filtering by using Eq. (1) to reject unrelated articles; and then we do tag removing, word segmentation and sentence splitting. There are two thresholds T_{num} and T_{filter} should be set in this parameterized approach for corpora constructing, T_{num} is about collected number of Wikipedia articles per term and T_{filter} is about information filtering. The bigger the collected number is, the more the related information we would obtain. However, it also brings with high cost of calculation, thus we should make a tradeoff. In this experiment, we empirically set T_{num} as 30 and T_{filter} as 10.0. The final corpora for "Database", "Operating System" and "Software Engineering" contain 1333 pages, 1656 pages and 1851 pages, respectively. After text processing, the three corpora contain 165,182 sentences, 212,600 sentences and 214,172 sentences, respectively.

4.1.2. *Learning process*

Relevance Calculation

After constructing one corpus for a taxonomy, we calculate semantic relatedness for every taxonomic concept. At the mean time, during the process of calculation, we retain the sentences with co-occurring concepts for further step of relation extraction.

Relation Extraction

From the calculated result of semantic relevance, we choose top concepts with the χ^2 value above a certain threshold of semantic relevance between concept pairs T_{conc} as the target objects. We apply the rules of dependency paring in the remained sentences containing co-occurring concepts. After obtaining the extracted predicates, we do statistical calculation again by using Eq. (2), and this time the statistical calculation objects are concept pairs and predicates.

Relation Generalization

By using Eq. (3) with the details described in Sec. 4.2.4, we adapt predicates to the more general level according to their domain and range. For instance, we adapt the object position of the relation "be-resident-in" from triple "loader be-resident-in main memory" to "loader be-resident-in memory", where "memory" is the super-class of "main memory" in "Operating System" ontology.

The following Tables 1, 2 and 3 show the top 10 χ^2-score of chosen predicates with concept pairs in the three ontologies.

Table 1. Top 10 χ^2-score of chosen predicates with concept pairs in "Database" taxonomy.

No.	Subject	Predicate	Object	χ^2	n_{11}	n_{10}	n_{01}	n_{00}
1.	database management system	organize data using	data model	5,175.59	1	63	0	165,618
2.	access method	retrieve record from	file	4,598.78	1	5	11	165,665
3.	data model	be organized into	tree	4,414.93	1	2	24	165,655
4.	access method	enable access to	data	1,183.59	3	61	36	165,582
5.	query language	manipulate	data	702.85	1	66	6	165,609
6.	file	contain	data	586.83	76	804	2478	162,324
7.	encoding scheme	used for	data compression	418.52	1	0	787	164,894
8.	data object	be stored in	array	351.03	1	1	467	165,213
9.	data model	constitute	database design	338.53	1	10	87	165,584
10.	database	provide	concurrency	178.48	11	56	2744	162,871

Table 2. Top 10 χ^2-score of chosen predicates with concept pairs in "Operating System" taxonomy.

No.	Subject	Predicate	Object	χ^2	n_{11}	n_{10}	n_{01}	n_{00}
1.	operating system	uses paging for	memory management	6,640.38	1	3	15	212,581
2.	operating system	need relocate	loader	4,425.9	1	31	2	212,566
3.	distributed system	be tracing	garbage collection	4,101.28	2	44	7	212,547
4.	stack	be allocated in	main memory	2,573.29	1	14	10	212,575
5.	thread	scheduled by	operating system	1,301.31	1	1	162	212,436
6.	loader	be resident in	memory	1,128.41	7	2	2014	210,577
7.	operating system	respond to	deadlock	531.52	1	2	264	212,333
8.	operating system	load	linker	234.11	4	5	3063	209,528
9.	linker	combine	file	191.37	1	2	727	211,870
10.	access control	provide	security	182.5	9	35	3652	208,904

Table 3. Top 10 χ^2-score of chosen predicates with concept pairs in "Software Engineering" taxonomy.

No.	Subject	Predicate	Object	χ^2	n_{11}	n_{10}	n_{01}	n_{00}
1.	assertion	provide a tool in	debugging	4,324.73	1	0	98	214,073
2.	model checking	be developed for checking	software design	1,805.37	1	0	236	213,935
3.	tracing	provide information for	debugging	1,204.61	1	0	354	213,817
4.	code reuse	stem from	structured programming	528.52	1	4	160	214,007
5.	assertion	be used to define	class invariant	477.42	1	2	296	213,873
6.	class invariant	can help	soft testing	152.9	1	2	913	213,256
7.	design pattern	be used with	object-oriented programming	135.47	2	12	855	213,303
8.	exception handling	easy	debugging	130.2	1	1	1607	212,563
9.	test automation	be used for	software testing	117.62	1	0	3580	210,591
10.	state diagram	be represented by	graph	115.74	1	4	717	213,450

4.1.3. *Program execution*

An IA-64 host with i7 CPU (3.4 G) and 20 G memory was used to execute the learning process after corpora construction. It takes less than 2 hours to complete the whole calculation on the parameter setting described in the beginning of Sec. 4.1.1.

4.2. *Evaluation*

As pointed out briefly in the introduction of this section, quantitative evaluation for non-taxonomic relations is difficult. The reason is because that the learning outcome of non-taxonomic relations is an open set. Given a specific taxonomy, understanding on its relational knowledge may vary among different knowledge engineers.

To customize some metrics for evaluating the non-taxonomic relations learning approach, we draw lessons from the research in [20]. Clearly, the centered element is the number of correct relations selected as final outcome, expressing real interactions between ordered concept pairs given. Assuming a gold standard exists, classic metrics in information retrieval like *precision*, *recall* and *F-measure* can be used for measuring the performance of learning approach. The *precision* (4) measures to which extend incorrect relations can be rejected. It is computed as the ratio between the number of correct relations selected ($n_{corr\text{-}selected}$) and the number of all relations extracted ($n_{all\text{-}extracted}$). The *recall* (5) measures to which correct relations can be accepted. It is computed as the ratio between the number of correctly selected relations ($n_{corr\text{-}selected}$) and the number of all relations in gold standard (n_{gold}). The *F-measure* (6) provides the weighted harmonic mean of precision and recall.

$$precision = \frac{n_{corr\text{-}selected}}{n_{all\text{-}extracted}} \tag{4}$$

$$recall = \frac{n_{corr\text{-}selected}}{n_{gold}} \tag{5}$$

$$F\text{-}measure = \frac{2 \times precision \times recall}{precision + recall} \tag{6}$$

However, as pointed out in the introduction of this section, the learning outcome of non-taxonomic relations is an open set. This makes it hard to find an available gold standard. Therefore, the *local_recall* (7) is adopted to cover such shortage. It is measured as the ratio between the number of correctly selected relations ($n_{corr\text{-}selected}$) against the number of correctly extracted relations ($n_{corr\text{-}extracted}$). If a high semantic threshold of semantic relevance between predicates and concept pair

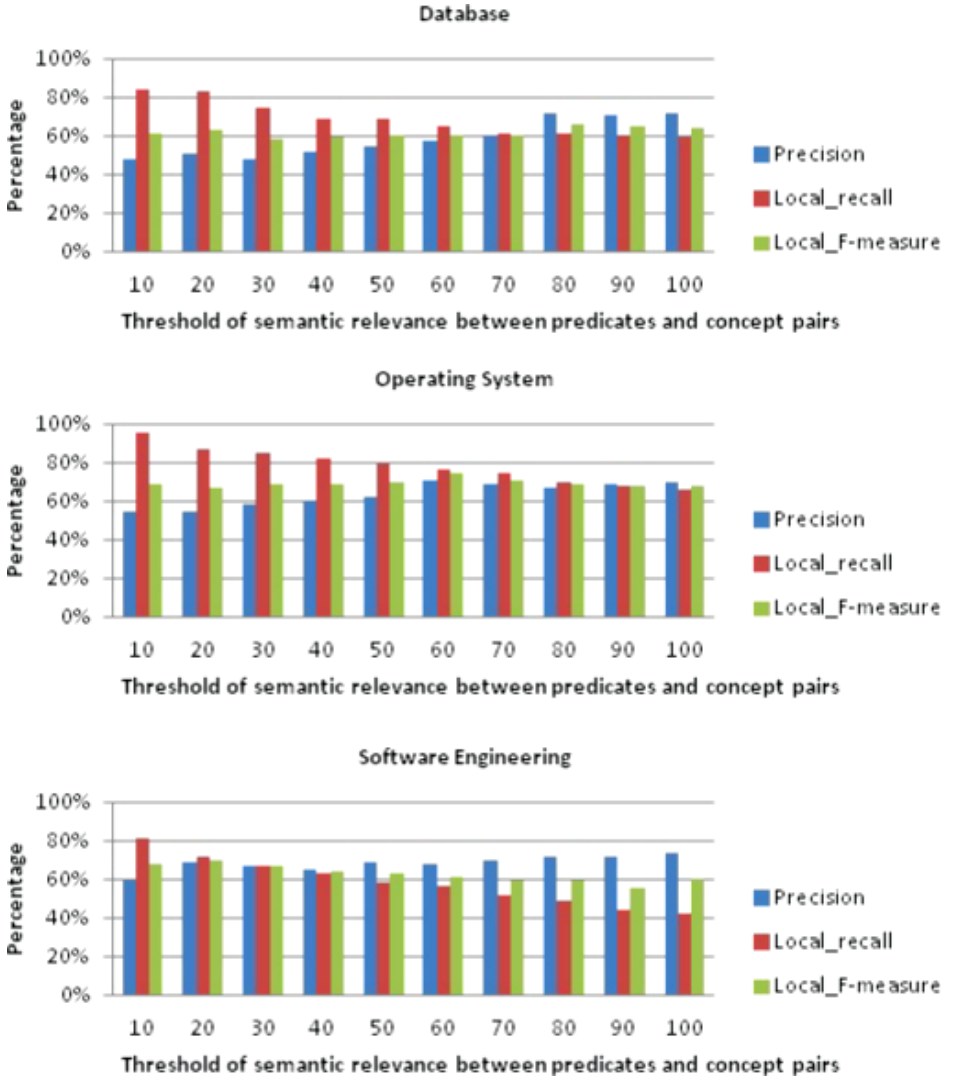

Fig. 4. Comparative analysis of non-taxonomic learning outcome for three taxonomies.

(T_{pred}) is used, some correct relations may be missed. Thus the *local_recall* can measure how well the learning approach is accepting or rejecting candidates. This metric is also used in [30, 31] on the situation without gold stand. In the same manner, *local_F-measure* (8) is used to replace the role of *F-measure*.

$$local\_recall = \frac{n_{corr\text{-}selected}}{n_{corr\text{-}extracted}} \tag{7}$$

$$local\_F\text{-}measure = \frac{2 \times precision \times local\_recall}{precision + local\_recall} \tag{8}$$

Evaluation is conducted by three experts of information science. By setting T_{pred} as 1.0, we obtain $n_{corr\text{-}extracted}$ for three ontologies as 83, 102 and 95, respectively. These values are set as the denominator of *local_recall*. This setting of T_{pred} is to prevent the noise that might be introduced through the small value of semantic relevance. Experts evaluate the *precision* (4), *local_recall* (7) and *local_F-measure* (8) under different setting of T_{pred}. Figure 4 shows the variation trend. It can be observed from the figure that, the *local_recall* is always reducing with the increasing of T_{pred}, while the precision is roughly increasing with the increasing of T_{pred}. The three taxonomies achieve best *local_F-measure* when T_{pred} is set as 80, 60, 20, respectively. All the three best *local_F-measure* are beyond 65%. For the sake of lack of gold standard, it is not likely to measure the true recall of non-taxonomic relations. For comparison, the referential work in [20] provides best *local_F-measure* in three experiments with 52%, 71%, 87%, which differentiate largely for different taxonomies. In [20], computation based on the statistical estimation of search engines may cause a certain deviation, as search engines conduct internal filtering on retrieval results. While our approach analyzes page information directly and thus provides more steady outputs.

5. Conclusions

In this paper, we propose a novel approach for learning non-taxonomic relations, which runs in a learning-on-demand mode. Given a taxonomy, it generate a learning corpus dynamically around taxonomic concepts, and explore possible arbitrary semantic relations within the taxonomy automatically by a three-step learning pipeline. The target taxonomy can be of any type meeting the demand of customers, which do not stick to domain boundaries. In fact, learning taxonomies automatically according to a certain document collection or a personalized requirement is a popular research direction currently [13–16]. Under these circumstances, the learned taxonomies may have blurred boundaries. They may focus a specific region of a domain or may span different domains. Faced with these diversified learned taxonomies, existing learning approaches based on static and domain-oriented corpora might not be flexible enough, while our approach can serve to extend taxonomies of these types.

Evaluation on three different taxonomies shows robustness and scalability of the proposed approach. The multiple filtering techniques based on statistical probability

contribute to the precision of learning results. In future research, class-instance detection can be introduced into the learning pipeline for improving the recall of learning results, as more instances would bring richer semantic information. In addition, another interesting research topic is to recognize different verb phrases with the same meanings, such as "be resident in" and "reside in".

Acknowledgments

We thank the anonymous reviewers for their valuable comments and suggestions. This research is supported by the National Natural Science Foundation of China under Grant Nos. 61232015 and 91318301.

References

1. T. R. Gruber, A translation approach to portable ontology specifications, *Knowledge Acquisition* **5**(2) (1993) 199–220.
2. S. Thaddeus and S. V. K. Raja, Ontology-driven model for knowledge-based software engineering, in *Proc. 18th Int. Conf. Software Engineering and Knowledge Engineering*, 2006, pp. 337–341.
3. M. Hecker, T. S. Dillon and E. Chang, Privacy ontology support for e-commerce, *Internet Computing* **12**(2) (2008) 54–61.
4. J. Saric, A. Gangemi, E. Ratsch and I. Rojas, Modelling gene expression, in *Proceedings of the Workshop on Models and Metaphors from Biology to Bioinformatics Tools*, 2004.
5. O. Arsene, I. Dumitrache and I. Mihu, Medicine expert system dynamic Bayesian network and ontology based, *Expert Systems with Applications* **38**(12) (2011) 15253–15261.
6. P. Cimiano, A. Mädche, S. Staab and J. Völker, Ontology learning, in *Handbook on Ontologies*, 2009, pp. 245–267.
7. A. D. Maedche, *Ontology Learning for the Semantic Web* (Kluwer Academic Publishers, 2002).
8. M. Ciaramita, A. Gangemi, E. Ratsch, J. Saric and I. Rojas, Unsupervised learning of semantic relations between concepts of a molecular biology ontology, in *Proc. IJCAI*, 2005.
9. A. Schutz and P. Buitelaar, Relext: A tool for relation extraction from text in ontology extension, *The Semantic Web*, 2005, pp. 593–606.
10. W. Wong, W. Liu and M. Bennamoun, Ontology Learning from text: A look back and into the future, *ACM Computing Surveys* **44**(4) (2012) 20.
11. J. D. Kim, T. Ohta, Y. Tateisi and J. I. Tsujii, GENIA corpus — a semantically annotated corpus for bio-textmining, *Bioinformatics* **19** (2003) i180–i182.
12. S. Pyysalo, F. Ginter, J. Heimonen, J. Bjorne, J. Boberg, J. Jarvinen and T. Salakoski, BioInfer: A corpus for information extraction in the biomedical domain, *BMC Bioinformatics* **8**(1) (2007) 50.
13. R. Navigli, P. Velardi and S. Faralli, A graph-based algorithm for inducing lexical taxonomies from scratch, in *Proc. 22nd Int. Joint Conf. on Artificial Intelligence*, 2011.
14. O. Medelyan, S. Manion, J. Broekstra, A. Divoli, A. L. Huang and I. H. Witten, Constructing a focused taxonomy from a document collection, in *Semantic Web: Semantics and Big Data*, 2013, pp. 367–381.

15. H. Yang and J. Callan, A metric-based framework for automatic taxonomy induction, in *Proc. Joint Conference of 47th Annual Meeting of ACL and 4th Int. Joint Conf. on Natural Language Processing of the AFNLP*, Vol. 1, 2009.
16. H. Yang, Constructing task-specific taxonomies for document collection browsing, in *Proc. Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, pp. 1278–1289.
17. A. D. Maedche and S. Staab, Discovering conceptual relations from text, ECAI, 2000, pp. 321–325.
18. M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead and O. Etzioni, Open information extraction from the Web, IJCAI (2007), pp. 2670–2676.
19. A. Fader, S. Soderland and O. Etzioni, Identifying relations for open information extraction, in *Proc. Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 1535–1545.
20. D. Sánchez and A. Moreno, Learning non-taxonomic relationships from web documents for domain ontology construction, *Data & Knowledge Engineering* **64**(3) (2008) 600–623.
21. W. Wong, W. Liu and M. Bennamoun, Acquiring semantic relations using the web for constructing lightweight ontologies, *Advances in Knowledge Discovery and Data Mining*, LNCS Vol. 5476, 2009, pp. 266–277.
22. Q. Liu, K. Xu, L. Zhang, H. Wang, Y. Yu and Y. Pan, Catriple: Extracting triples from Wikipedia categories, *The Semantic Web*, 2008, pp. 330–344.
23. T. P. Mohamed, E. R. Hruschka Jr and T. M. Mitchell, Discovering relations between noun categories, in *Proc. Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 1447–1455.
24. I. Serra, R. Girardi and P. Novais, The problem of learning non-taxonomic relationships of ontologies from text, in *Distributed Computing and Artificial Intelligence*, 2012, pp. 485–492.
25. C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing* (MIT Press, 1999).
26. M. Marneffe, B. MacCartney and C. Manning, Generating typed dependency parses from phrase structure parses, in *Proc. LREC*, 2006, pp. 449–454.
27. D. Hindle, Noun classification from predicate-argument structures, in *Proc. of 28th Annual Meeting on Association for Computational Linguistics*, 1990, pp. 268–275.
28. P. Cimiano, M. Hartung and E. Ratsch, Finding the appropriate generalization level for binary ontological relations extracted from the Genia corpus, in *Proc. of Int. Conf. on Language Resources and Evaluation*, 2006, pp. 161–169.
29. K. Dellschaft and S. Steffen, Strategies for the evaluation of ontology learning, in *Proc. Conf. Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, 2008, pp. 253–272.
30. E. Alfonseca and S. Manandhar, An unsupervised method for general named entity recognition and automated concept discovery, in *Proc. of First International Conference on General WordNet*, 2002.
31. O. Etzioni *et al.*, Unsupervised named-entity extraction from the Web: An experimental study, *Artificial Intelligence* **165** (2005) 91–134.