Taxonomy Construction Using Syntactic Contextual Evidence

Luu Anh Tuan #1, **Jung-jae Kim** #2, **Ng See Kiong** *3

#*School of Computer Engineering, Nanyang Technological University, Singapore*

1anhtuan001@e.ntu.edu.sg, 2jungjae.kim@ntu.edu.sg

**Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore*

3skng@i2r.a-star.edu.sg

Abstract

Taxonomies are the backbone of many structured, semantic knowledge resources. Recent works for extracting taxonomic relations from text focused on collecting lexical-syntactic patterns to extract the taxonomic relations by matching the patterns to text. These approaches, however, often show low coverage due to the lack of contextual analysis across sentences. To address this issue, we propose a novel approach that collectively utilizes contextual information of terms in syntactic structures such that if the set of contexts of a term includes most of contexts of another term, a subsumption relation between the two terms is inferred. We apply this method to the task of taxonomy construction from scratch, where we introduce another novel graph-based algorithm for taxonomic structure induction. Our experiment results show that the proposed method is well complementary with previous methods of linguistic pattern matching and significantly improves recall and thus F-measure.

1 Introduction

Taxonomies that are backbone of structured ontology knowledge have been found to be useful for many areas such as question answering (Harabagiu et al., 2003), document clustering (Fodeh et al., 2011) and textual entailment (Geffet and Dagan, 2005). There have been an increasing number of hand-crafted, well-structured taxonomies publicly available, including WordNet (Miller, 1995), OpenCyc (Matuszek et al., 2006), and Freebase (Bollacker et al., 2008). However, the manual curation of those taxonomies is time-consuming and human experts may miss relevant terms. As such, there are still needs to extend existing taxonomies or even to construct new taxonomies from scratch.

The previous methods for identifying taxonomic relations (i.e. is-a relations) from text can be generally classified into two categories: statistical and linguistic approaches. The former includes co-occurrence analysis (Budanitsky, 1999), term subsumption (Fotzo and Gallinari, 2004) and clustering (Wong et al., 2007). The main idea behinds these techniques is that the terms that frequently co-occur may have taxonomic relationships. Such approaches, however, usually suffer from low accuracy, though relatively high coverage, and heavily depend on the choice of feature types and datasets. Most previous methods of the linguistic approach, on the other hand, rely on the lexical-syntactic patterns (e.g. *A is a B*, *A such as B*) (Hearst, 1992). Those patterns can be manually created (Kozareva et al., 2008; Wentao et al., 2012), chosen via automatic bootstrapping (Widdows and Dorow, 2002; Girju et al., 2003) or identified from machine-learned classifiers (Navigli et al., 2011). The pattern matching methods generally achieve high precision, but low coverage due to the lack of contextual analysis across sentences. In this paper, we introduce a novel statistical method and shows that when combined with a pattern matching method, it shows significant performance improvement.

The proposed statistical method, called syntactic contextual subsumption (SCS), compares the syntactic contexts of terms for the taxonomic relation identification, instead of the usage of bag-of-words model by the previous statistical methods. We observe that the terms in taxonomic relations may not occur in the same sentences, but in similar syntactic structures of different sentences, and that the contexts of a specific term are often found in the contexts of a general term but not vice versa. By context of a term, we mean the set of words frequently have a particular syntactic relation (e.g. *Subject-Verb-Object*) with the term in a given corpus. Given two terms, the SCS method collects from the Web pre-defined syntactic relations of each of the terms and checks if the syntactic contexts of a term properly includes that of the other term in order to determine their taxonomic relation. The method scores each taxonomic relation candidate based on the two measures of Web-based evidence and contextual set inclusion, and as such, is able to find implicit subsumption relations between terms across sentences. The SCS shows itself (Section 3.1) to be complementary to linguistic pattern matching.

After the relation identification, the identified taxonomic relations should be integrated into a graph for the task of taxonomy construction from scratch or associated with existing concepts of a given taxonomy via is-a relations (Snow et al., 2006). In this step of taxonomic structure construction, there is a need for pruning incorrect and redundant relations. Previous methods for the pruning task (Kozareva and Hovy, 2010; Velardi et al., 2012) treat the identified taxonomic relations equally, and the pruning task is thus reduced to finding the best trade-off between path length and the connectivity of traversed nodes. This assumption, however, is not always true due to the fact that the identified taxonomic relations may have different confidence values, and the relations with high confidence values can be incorrectly eliminated during the pruning process. We thus propose a novel method for the taxonomy induction by utilizing the evidence scores from the relation identification method and the topological properties of the graph. We show that it can effectively prune redundant edges and remove loops while preserving the correct edges of taxonomy.

We apply the proposed methods of taxonomic relation identification and taxonomy induction to the task of constructing a taxonomy from a given text collection from scratch. The resultant system consists of three modules: Term extraction and filtering (Section 2.1), taxonomic relation identification (Section 2.2), and taxonomy induction (Section 2.3). The outputs of the term extraction/filtering module are used as inputs of the taxonomic relation identification, such that the taxonomic relation identification module checks if there is a taxonomic relation between each pair of terms from the term extraction/filtering module. The taxonomy induction module gets the identified taxonomic relation set as the input, and outputs the final optimal taxonomy by pruning redundant and incorrect relations.

2 Methodology

2.1 Term Extraction and Filtering

The first step to construct taxonomies is to collect candidate terms from text documents in the domain of interest. Like most of linguistic approaches, we use pre-defined linguistic filters to extract candidate terms, including single-word terms and multi-word terms which are noun or noun phrases in sentences. These terms are then preprocessed by removing determiners and lemmatization.

The candidate terms collected are then filtered to select the terms that are most relevant to the domain of interest. Many statistical techniques are developed for the filtering, such as $TF\text{-}IDF$, domain relevance (DR), and domain consensus (DC) (Navigli and Velardi, 2004). DR measures the amount of information that a term t captures within a domain of interest D_i, compared to other contrasting domains (D_j), while DC measures the distributed use of a term t across documents d in a domain D_i. Since three measures have pros and cons, and might be complementary to each other, our term filtering method is thus the linear combination of them:

$$\begin{aligned} TS(t, D_i) = {} & \alpha \times TFIDF(t, D_i) \\ & + \beta \times DR(t, D_i) + \gamma \times DC(t, D_i) \end{aligned} \tag{1}$$

We experimented (see Section 3) with different values of α, β and γ, and found that the method shows the best performance when the values for α and β are 0.2 and 0.8 and the value for γ is between 0.15 and 0.35, depending on the size of the domain corpus.

2.2 Taxonomic Relation Identification

In this section, we present three taxonomic relation identification methods which are adopted in our system. First, two methods of string inclusion with WordNet and lexical-syntactic pattern matching, which were commonly used in the literature will be introduced with some modifications. Then, a novel syntactic contextual subsumption method to find implicit relations between terms across sentences by using contextual evidence from syntactic structures and Web data will be proposed. Finally, these three methods will be linearly combined to

Notation	Meaning
$t_1 \gg t_2$	t_1 is a hypernym of t_2
$t_1 \approx t_2$	t_1 semantically equals or is similar to t_2
$t_1 \gg_{WN} t_2$	t_1 is a direct or inherited hypernym of t_2 according to WordNet
$t_1 \approx_{WN} t_2$	t_1 and t_2 belong to the same synset of WordNet

Table 1: Notations

form an integrating solution for taxonomic relation identification. Given two terms t_1 and t_2, Table 1 summarizes important notations used in this paper.

2.2.1 String Inclusion with WordNet (SIWN)

One simple way to check taxonomic relation is to test string inclusion. For example, "terrorist organization" is a hypernym of "foreign terrorist organization", as the former is a substring of the latter. We propose an algorithm to extend the string inclusion test by using WordNet, which will be named SIWN. Given a candidate general term t_g and a candidate specific term t_s, the SIWN algorithm examines t_g from left to right (designating each word in t_g to be examined as w_g) to check if there is any word (w_s) in t_s such that $w_g \approx_{WN} w_s$ or $w_g \gg_{WN} w_s$, and identifies the taxonomic relation between two terms if every word of t_g has a corresponding word in t_s (with at least one $\gg_{WN}$ relation). For example, consider two terms: "suicide attack" and "world trade center self-destruction bombing". Because "attack" $\gg_{WN}$ "bombing" and "suicide" $\approx_{WN}$ "self-destruction", according to SIWN algorithm, we conclude that "suicide attack" is the hypernym of "world trade center self-destruction bombing".

Given two terms t_1 and t_2, the evidence score for SIWN algorithm is calculated as follows:

$$Score_{SIWN}(t_1, t_2) = \begin{cases} 1 & \text{if } t1 \gg t_2 \text{ via SIWN} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

2.2.2 Lexical-syntactic Pattern

Extending the ideas of Kozareva and Hovy (2010) and Navigli et al. (2011), we propose a method of extracting taxonomic relations by matching lexical-syntactic patterns to the Web data.

Definition 1 (Syntactic patterns). *Given two terms t_1 and t_2, Pat(t_1, t_2) is defined as the set of the following patterns:*

- *"t_1 such as t_2"*
- *"t_1, including t_2"*
- *"t_2 is [a|an] t_1"*
- *"t_2 is a [kind|type] of t_1"*
- *"t_2, [and|or] other t_1"*

, where t_1 and t_2 are replaced with actual terms and [a|b] denotes a choice between a and b.

Given candidate general term t_1 and candidate specific term t_2, the lexical-syntactic pattern (LSP) method works as follows:

1. Submit each phrase in *Pat*(t_1, t_2) to a Web search engine as a query. The number of the search results of the query is denoted as *WH*(t_1, t_2).
2. Calculate the following evidence score:
$$Score_{LSP}(t_1, t_2) = \frac{\log(WH(t_1, t_2))}{1 + \log(WH(t_2, t_1))} \quad (3)$$
3. If $Score_{LSP}(t_1, t_2)$ is greater than a threshold value then $t_1 \gg t_2$.

While most lexical-syntactic pattern methods in the literature only consider the value of *WH*(t_1, t_2) in checking $t_1 \gg t_2$ (Wentao et al., 2012), we take into account both *WH*(t_1, t_2) and *WH*(t_2, t_1). The intuition of formula (3) is that if $t1$ is a hypernym of $t2$ then the size of *WH*(t_1, t_2) will be much larger than that of *WH*(t_2, t_1), which means the lexical-syntactic patterns are more applicable for the ordered pair (t_1, t_2) than (t_2, t_1).

2.2.3 Syntactic Contextual Subsumption

The LSP method performs well in recognizing the taxonomic relations between terms in the sentences containing those pre-defined syntactic patterns. This method, however, has a major shortcoming: it cannot derive taxonomic relations between two terms occurring in two different sentences. We thus propose a novel syntactic contextual subsumption (SCS) method which utilizes contextual information of terms in syntactic structure (i.e. *Subject-Verb-Object* in this study) and Web data to infer implicit taxonomic relations between terms across sentences. Note that the chosen syntactic structure *Subject-Verb-Object* is identical to the definition of non-taxonomic relations in the literature (Buitelaar et al., 2004), where the *Verb* indicate non-taxonomic relations between *Subject* and *Object*. In this subsection, we first present the method to collect those non-taxonomic relations. Then we present in detail the ideas of the SCS method and how we can use it to derive taxonomic relations in practice.

A. Non-taxonomic Relation Identification

Following previous approaches to non-taxonomic relation identification, e.g. (Ciaramita et al., 2005), we use the Stanford parser (Klein and Manning, 2003) to identify the syntactic structures of sentences and extract triples of *(Subject, Verb, Object)*, where *Subject* and *Object* are noun phrases.

We further consider the following issues: First, if a term (or noun phrase) includes a preposition, we remove the prepositional phrase. However, if the headword of a term is a quantitative noun like "lot", "many" or "dozen" and it is modified by the preposition "of", we replace it with the headword of the object of the preposition "of". For example, we can extract the triples $(people, need, food)$ and $(people, like, snow)$ from the following sentences, respectively:

- "People in poor countries need food"
- "A lot of people like snow"

Second, if the object of a verb is in a verb form, we replace it with, if any, the object of the embedded verb. For example, we can extract the triple $(soldier, attack, terrorist)$ from the following sentence:

- "The soldiers continue to attack terrorists"

Third, if a term has a coordinate structure with a conjunction like "and" or "or", we split it into all coordinated noun phrases and duplicate the triple by replacing the term with each of the coordinated noun phrases. For example, we can extract the triples of $R(girl, like, dog)$ and $R(girl, like, cat)$ from the following sentence:

- "The girl likes both dogs and cats"

Given two terms t_1, t_2 and a non-taxonomic relation r, some notations which will be used hereafter are shown below:

- $R(t_1, r, t_2)$: t_1, r, and t_2 have a *(Subject, Verb, Object)* triple.
- $\Theta(t_1, t_2)$: the set of relations r such that there exists $R(t_1, r, t_2)$ or $R(t_2, r, t_1)$.

B. Syntactic Contextual Subsumption Method

The idea of the SCS method derived from the following two observations.

Observation 1. *Given three terms t_1, t_2, t_3, and a non-taxonomic relation r, if we have two triples $R(t_1, r, t_3)$ and $R(t_2, r, t_3)$ (or $R(t_3, r, t_1)$ and $R(t_3, r, t_2)$), t_1 and t_2 may be in taxonomic relation.*

For example, given two triples *R(Al-Qaeda, attack, American)* and *R(Terrorist group, attack, American)*, a taxonomic relation *Terrorist group* $\gg$ *Al-Qaeda* can be induced. However, it is not always guaranteed to induce a taxonomic relations from such a pair of triples, for example from *R(animal, eat, meat)* and *R(animal, eat, grass)*. The second observation introduced hereafter will provide more chance to infer taxonomic relationship.

Definition 2 (Contextual set of a term)**.** *Given a term t_1 and a non-taxonomic relation r, $S(t_1, r, \text{"}subj\text{"})$ denotes the set of terms t_2 such that there exists triple $R(t_1, r, t_2)$. Similarly, $S(t_1, r, \text{"}obj\text{"})$ is the set of terms t_2 such that there exists triple $R(t_2, r, t_1)$.*

Observation 2. *Given two terms t_1, t_2, and a non-taxonomic relation r, if $S(t_1, r, \text{"}subj\text{"})$ mostly contains $S(t_2, r, \text{"}subj\text{"})$ but not vice versa, then most likely t_1 is a hypernym of t_2. Similarly, if $S(t_1, r, \text{"}obj\text{"})$ mostly contains $S(t_2, r, \text{'}obj\text{"})$ but not vice versa, then most likely t_1 is a hypernym of t_2.*

For example, assume that *S(animal, eat, "subj")* = $\{$*grass, potato, mouse, insects, meat, wild boar, deer, buffalo*$\}$ and *S(tiger, eat, "subj")* = $\{$*meat, wild boar, deer, buffalo*$\}$. Since *S(animal, eat, "subj")* properly contains *S(tiger, eat, "subj")*, we can induce *animal* $\gg$ *tiger*.

Based on Observation 2, our strategy to infer taxonomic relations is to first find the contextual set of terms via the evidence of syntactic structures and Web data, and then compute the score of the set inclusion. The detail of the method is presented hereafter.

Definition 3. *Given two terms t_1, t_2 and a non-taxonomic relation r, $C(t_1, t_2, r, \text{``}subj\text{''})$ denotes the number of terms $t_3$ such that there exists both triples $R(t_1, r, t_3)$ and $R(t_2, r, t_3)$. Similarly, $C(t_1, t_2, r, \text{``}obj\text{''})$ is the number of terms t_3 such that there exists both relations $R(t_3, r, t_1)$ and $R(t_3, r, t_2)$.*

Given the pair of a candidate general term t_1 and a candidate specific term t_2, we extract their non-taxonomic relations from corpora extracted from the Web, and use them to determine the taxonomic relation between t_1 and t_2 as follows:

1. Find from a domain corpus the relation r and type Γ such that:
$$C(t_1, t_2, r, \Gamma) = \max_{\substack{r' \in \Theta(t_1, t_2) \\ \Gamma' \in \{\text{``}subj\text{''}, \text{``}obj\text{''}\}}} C(t_1, t_2, r', \Gamma')$$

2. If type Γ is "subj", collect the first 1,000 search results of the query "t_1 r" using the Google search engine, designated as $Corpus^{\Gamma}_{t_1}$. In the same way, construct $Corpus^{\Gamma}_{t_2}$ with the query "t_2 r". If Γ is "obj", two queries "r t_1" and "r t_2" are submitted instead to collect $Corpus^{\Gamma}_{t_1}$ and $Corpus^{\Gamma}_{t_2}$, respectively.

3. Find the sets of $S(t_1, r, \Gamma)$ and $S(t_2, r, \Gamma)$ from $Corpus^{\Gamma}_{t_1}$ and $Corpus^{\Gamma}_{t_2}$, respectively, using the non-taxonomic relation identification method above.

4. Calculate the following evidence score for SCS method:
$$Score_{SCS} = \left[\frac{|S(t_1, r, \Gamma) \bigcap S(t_2, r, \Gamma)|}{|S(t_2, r, \Gamma)|} + \left(1 - \frac{|S(t_1, r, \Gamma) \bigcap S(t_2, r, \Gamma)|}{|S(t_1, r, \Gamma)|}\right) \right] \times \log(|S(t_1, r, \Gamma)| + |S(t_2, r, \Gamma)|) \quad (4)$$
The basic idea of the contextual subsumption score in our method is that if t_1 is a hypernym of t_2 then the set $S(t_1, r, \Gamma)$ will mostly contain $S(t_2, r, \Gamma)$ but not vice versa. The intuition of formula (5) is inspired by Jaccard similarity coefficient. We then multiply the score with the log value of total size of two sets to avoid the bias of small set inclusion.

5. If $Score_{SCS}(t_1, t_2)$ is greater than a threshold value, then we have $t1 \gg t2$.

2.2.4 Combined Method

In our study, we linearly combine three methods as follows:

1. For each ordered pair of terms (t_1, t_2) calculate the total evidence score:
$$\begin{aligned} Score(t_1, t_2) = \; & \alpha \times Score_{SIWN}(t_1, t_2) \\ & + \beta \times Score_{LSP}(t_1, t_2) \\ & + \gamma \times Score_{SCS}(t_1, t_2) \end{aligned} \quad (5)$$

2. If $Score(t_1, t_2)$ is greater than a threshold value, then we have $t_1 \gg t_2$.

We experimented with various combinations of values for α, β and γ, and found that the method shows the best performance when the value of α is 0.5, β is between 0.35 and 0.45, and γ is between 0.15 and 0.25, depending on the domain corpus size.

2.3 Taxonomy Induction

The output of the taxonomic relation identification module is a set of taxonomic relations T. In this section, we will introduce a graph-based algorithm (Algorithm 1) to convert this set into an optimal tree-structured taxonomy, as well as to eliminate incorrect and redundant relations. Denote $e(t_1, t_2)$ as an directed edge from t_1 to t_2, the algorithm consists of three steps which will be described hereafter with the corresponding lines in Algorithm 1.

Step 1: Initial hypernym graph creation (line 1 - 16) This step is to construct a connected directed graph from the list of taxonomic relations. The idea is to add each taxonomic relation $t_1 \gg t_2$ as a directed edge from parent node t_1 to child node t_2, and if t_1 does not have any hypernym term, t_1 will become a child node of $ROOT$ node. The result of this step is a connected graph containing all taxonomic relations with the common $ROOT$ node.

Step 2: Edge weighting (line 17) This step is to calculate the weight of each edge in the hypernym graph. Unlike the algorithm of Velardi et al. (2012) and Kozareva and Hovy (2010) where every taxonomic relation is treated equally, we assume the confidence of each taxonomic relation is different, depending on the amount of

Algorithm 1 Taxonomy Induction Algorithm

```
Input:  $T$ : the taxonomic relation set
Output: $V$: the vertex set of resultant taxonomy;
        $E$: the edge set of resultant taxonomy;
 1: Initialize $V = \{ROOT\}$, $E = \emptyset$;
 2: for each taxonomic relation $(t_1 \gg t_2) \in T$ do
 3:     $E = E \cup \{e(t_1, t_2)\}$
 4:     if $t_1 \notin V$ then
 5:         $V = V \cup \{t_1\}$
 6:     end if
 7:     if $t_2 \notin V$ then
 8:         $V = V \cup \{t_2\}$
 9:     end if
10:     if $\nexists\, e(t_3, t_1) \in E$ with $t_3 \neq ROOT$ then
11:         $E = E \cup \{e(ROOT, t_1)\}$
12:     end if
13:     if $\exists\, e(ROOT, t_2) \in E$ then
14:         $E = E \setminus \{e(ROOT, t_2)\}$
15:     end if
16: end for
17: edgeWeighting($V, E$);
18: graphPruning($V, E$);
```

evidence it has. Thus, the hypernym graph edges will be weighted as follows:

$$w(e(t_1, t_2)) = \begin{cases} 1 & \text{if } t_1 = ROOT \\ Score(t_1, t_2) & \text{otherwise} \end{cases} \quad (6)$$

Note that the $Score$ value in formula (6) is determined by the taxonomic relation identification process described in Section 2.2.4.

Step 3: Graph pruning (line 18) The hypernym graph generated in Step 1 is not an optimal taxonomy as it may contain many redundant edges or incorrect edges which together form in a loop. In this step, we aim at producing an optimal taxonomy by pruning the graph based on our edge weighting strategy. A maximum spanning tree algorithm, however, cannot be applied as the graph is directed. For this purpose, we apply Edmonds' algorithm (Edmonds, 1967) for finding a maximum optimum branching of a weighted directed graph. Using this algorithm, we can find a subset of the current edge set, which is the optimized taxonomy where every non-root node has in-degree 1 and the sum of the edge weights is maximized. Figure 1 shows an example of the taxonomy induction process.

3 Experiment Results

We evaluated our methods for taxonomy construction against the following text collections of five domains:

- Artificial Intelligence (AI) domain: 4,119 papers extracted from the IJCAI proceedings from 1969 to 2011 and the ACL archives from year 1979 to 2010. The same dataset used in the work of Velardi et al. (2012).
- Terrorism domain: 104 reports of the US state department, titled "Patterns of Global Terrorism (1991-2002)" [1]. A report contains about 1,500 words.
- Animals, Plants and Vehicles domains: Collections of Web pages crawled by using the bootstrapping algorithm described by Kozareva et al. (2008). Navigli et al. (2011) and Kozareva and Hovy (2010) used these datasets to compare their outputs against WordNet sub-hierarchies.

There are two experiments performed in this section: 1) Evaluating the construction of new taxonomies for Terrorism and AI domains, and 2) Comparing our results with the gold-standard WordNet sub-hierarchies. Note that in the experiments, the threshold value we used for $Score_{LSP}$ is 1.9, $Score_{SCS}$ is 1.5 and $Score$ is 2.1.

3.1 Constructing new taxonomies for AI and Terrorism domains

Referential taxonomy structures such as WordNet or OpenCyc are widely used in semantic analytics applications. However, their coverage is limited to common well-known areas, and many specific domains like Terrorism and AI are not well covered in those structures. Therefore, an automatic method which can induce taxonomies for those specific domains from scratch can greatly contribute to the process of knowledge discovery.

First, we applied our taxonomy construction system to the AI domain corpus. We compared the taxonomy constructed by our system with that obtained by Velardi et al. (2012), and show the comparison results in Table 2. Notice that in this comparison, to be fair, we use the same set of terms that was used in (Velardi et al., 2012). The result shows that our approach can extract 9.8%

[1] *http://www.fas.org/irp/threat/terror.htm*

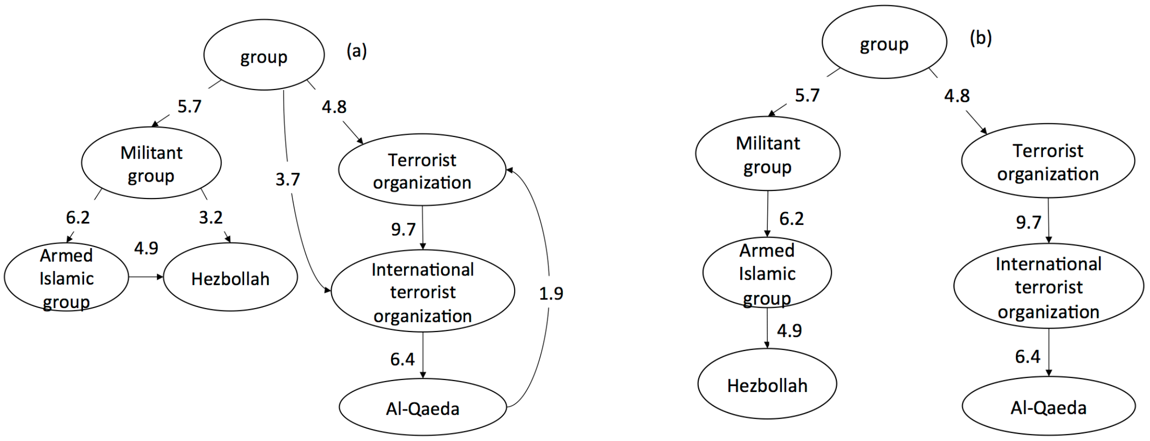

Figure 1: An example of taxonomy induction. (a) Initial weighted hypernym graph. (b) Final optimal taxonomy, where we prune two redundant edges *(group, International terrorist organization)*, *(Militant group, Hezbollah)* and remove the loop by cutting an incorrect edge *(Al-Qaeda, Terrorist organization)*.

more taxonomic relations and achieve 7% better term coverage than Velardi's approach.

	Our system	Velardi's system
#vertex	1839	1675
#edge	1838	1674
Average depth	6.2	6
Max depth	10	10
Term coverage	83%	76%

Table 2: Comparison of our system with (Velardi et al., 2012)

We also applied our system to the Terrorism corpus. The proposed taxonomic relation identification algorithm extracts a total of 976 taxonomic relations, from which the taxonomy induction algorithm builds the optimal taxonomy. The total number of vertices in the taxonomy is 281, and the total number of edges is 280. The average depth of the trees is 3.1, with the maximum depth 6. In addition, term coverage (the ratio of the number of terms in the final optimal trees to the number of terms obtained by the term suggestion/filtering method) is 85%.

To judge the contribution of each of taxonomic relation identification methods described in Section 2.2 to the overall system, we alternately run the system for the AI and Terrorism domains with different combinations of the three methods (i.e. SIWN, LSP, and SCS) as shown in Table 3. Note that we employed only the first two modules of term suggestion/filtering and taxonomic relation identification except the last module of taxonomy induction for this experiment. Table 3 shows the number of the taxonomic relations extracted by each of the combinations. Since SIWN and LSP are commonly used by previous taxonomic relation identification systems, we consider the combination of SIWN + LSP as the baseline of the experiment. The results in Table 3 show that the three methods are all well complementary to each other. In addition, the proposed SCS method can contribute up to about 27% - 29% of all the identified taxonomic relations, which were not discovered by the other two baseline methods.

	No. of extracted relations	
	Terrorism	AI domain
SCS	484	1308
SIWN	301	984
LSP	527	1537
SIWN + LSP	711	2203
SCS + SIWN + LSP	976	3122

Table 3: The number of taxonomic relations extracted by different methods.

	Percentage of correct relations	
	Terrorism	AI domain
SCS	91%	88%
SIWN	96%	91%
LSP	93%	93%
SCS + SIWN + LSP	92%	90%

Table 4: Estimated precision of taxonomic relation identification methods in 100 extracted relations.

	Animals domain			Plants domain			Vehicles domain		
	Our	Kozareva	Navigli	Our	Kozareva	Navigli	Our	Kozareva	Navigli
#Correct relations	2427	1643	N.A.	1243	905	N.A.	281	246	N.A.
Term coverage	96%	N.A.	94%	98%	N.A.	97%	97%	N.A.	96%
Precision	95%	98%	97%	95%	97%	97%	93%	99%	91%
Recall	56%	38%	44%	53%	39%	38%	69%	60%	49%
F-measure	71%	55%	61%	68%	56%	55%	79%	75%	64%

Table 5: Comparison of (Navigli et al., 2011), (Kozareva and Hovy, 2010) and our system against WordNet in three domains: Animals, Plants and Vehicles.

We further evaluated the precision of each individual taxonomic relation identification method. For AI and Terrorism domains, we again run the system with each of the three methods and with all together, and then randomly select 100 extracted taxonomic relations each time. These selected taxonomic relations are then examined by two domain experts to check the correctness. The evaluation results are given in Table 4. Note that only the first two modules of term suggestion/filtering and taxonomic relation identification are employed for this experiment as well. The SIWN and LSP methods achieve high precision because they are based on the gold-standard taxonomy hierarchy WordNet and on the well-defined patterns, respectively. In contrast, the SCS method ambitiously looks for terms pairs that share similar syntactic contexts across sentences, though the contextual evidence is restricted to certain syntactic structures, and thus has a slightly lower precision compared to the other two methods.

In short, the SCS method is complementary to the baseline methods, significantly improving the coverage of the combined methods, when its precision is comparable to those of the baseline methods. We performed next experiments to show that the SCS method overall has synergistic impact to improve the F-measure of the combined methods.

3.2 Evaluation against WordNet

In this experiment, we constructed taxonomies for three domains Animals, Plants and Vehicles, and then checked whether the identified relations can be found in the WordNet, and which relations in WordNet are not found by our method. Note that in this comparison, to be fair, we changed our algorithm to avoid using WordNet in identifying taxonomic relations. Specifically, in the SIWN algorithm, all operations of "$\approx_{WN}$" are replaced with normal string-matching comparison, and all "$\gg_{WN}$" relations are falsified. The evaluation uses the following measures:

$$Precision = \frac{\#relations\ found\ in\ WordNet\ and\ by\ the\ method}{\#relations\ found\ by\ the\ method}$$

$$Recall = \frac{\#relations\ found\ in\ WordNet\ and\ by\ the\ method}{\#relations\ found\ in\ WordNet}$$

We also compared our results with those obtained by the approaches of Navigli et al. (2011) and Kozareva and Hovy (2010), where they also compared their resultant taxonomies against WordNet. In this comparison, all the three approaches (i.e. ours, the two previous methods) use the same corpora and term lists. The comparison results are given in Table 5. "$N.A.$" value means that this parameter is not applicable to the corresponding method. The results show that our approach achieves better performance than the other two approaches, in terms of both the number of correctly extracted taxonomic relations and the term coverage. Our system has a slightly lower precision than that of (Navigli et al., 2011) and (Kozareva and Hovy, 2010) due to the SCS method, but it significantly contributes to improve the recall and eventually the F-measure over the other two systems.

To judge the effectiveness of our proposed taxonomy induction algorithm described in Section 2.3, we compared it with the graph-based algorithm of Velardi et al. (2012). Recall that in this algorithm, they treat all taxonomic relations equally, and the pruning task is reduced to finding the best trade-off between path length and the connectivity of traversed nodes. For each of five domains (i.e. Terrorism, AI, Animals, Plants and Vehicles), we alternately run the two taxonomy induction algorithms over the same taxonomic relation set produced by our taxonomic relation identification process. For Terrorism and AI domains, we randomly pick up 100 edges in each resultant taxonomy and ask two domain experts to judge for the correctness. For Animals, Plants and Vehicles domains, we check the correctness of the edges in resultant taxonomies by comparing them against the corresponding sub-hierarchies in WordNet. The evaluation is given in Table 6. The results show that the proposed taxonomy induction algorithm can achieve better performance than the algorithm of Velardi et al. (2012). This may be due to the fact that our algorithm considers the scores of the identified taxonomic relations from the relation identification module, and thus is more precise in eliminating incorrect relations during the pruning process.

	Percentage of correct edges	
	Our algorithm	Velardi's algorithm
Terrorism	94%	90%
AI	93%	88%
Animals	95%	93%
Plants	95%	92%
Vehicles	93%	92%

Table 6: Comparison of our taxonomy induction algorithms and that of Velardi et al. (2012).

In addition, when comparing Tables 4 and 6, we can find that the precision of taxonomic relations after the pruning process is higher than that before the pruning process, which proves that the proposed taxonomy induction algorithm effectively trims the incorrect relations of Terrorism and AI taxonomies, leveraging the percentage of correct relations 2% - 3% up.

For the SCS method, besides the triple *Subject-Verb-Object*, we also explore other syntactic structures like *Noun-Preposition-Noun* and *Noun-Adjective-Noun*. For example, from the sentence *"I visited Microsoft in Washington"*, the triple *(Microsoft, in, Washington)* is extracted using *Noun-Preposition-Noun* structure. Similarly, from the sentence *"Washington is a beautiful city"*, the triple *(Washington, beautiful, city)* is extracted using *Noun-Adjective-Noun* structure. We then use the triples for the contextual subsumption method described in Section 2.2.3, and test the method against the Animals, Plants and Vehicles domains. The results are then compared against WordNet sub hierarchies. The experiment results in Table 7 show that the triples of *Subject-Verb-Object* give the best performance compared to the other syntactic structures. These can be explained as the number of triples of two types *Noun-Preposition-Noun* and *Noun-Adjective-Noun* are smaller than that of *Subject-Verb-Object*, and the number of *Verb* is much greater than number of *Preposition* or *Adjective*.

	S-V-O	*N-P-N*	*N-A-N*
Animals domain			
Precision	95%	68%	72%
Recall	56%	52%	47%
F-measure	71%	59%	57%
Plants domain			
Precision	95%	63%	66%
Recall	53%	41%	43%
F-measure	68%	50%	52%
Vehicles domain			
Precision	93%	59%	60%
Recall	69%	45%	48%
F-measure	79%	51%	53%

Table 7: Comparison of three syntactic structures: *S-V-O* (*Subject-Verb-Object*), *N-P-N* (*Noun-Preposition-Noun*) and *N-A-N* (*Noun-Adjective-Noun*).

All experiment results are available at *http://nlp.sce.ntu.edu.sg/wiki/projects/taxogen*.

4 Conclusion

In this paper, we proposed a novel method of identifying taxonomic relations using contextual evidence from syntactic structure and Web data. This method is proved well complementary with previous method of linguistic pattern matching. We also present a novel graph-based algorithm to induce an optimal taxonomy from a given taxonomic relation set. The experiment results show that our system can generally achieve better performance than the state-of-the-art methods. In the future, we will apply the proposed taxonomy construction method to other domains such as biomedicine and integrate it into other frameworks such as ontology authoring.

References

K. Bollacker, C. Evans, P. Paritosh, T. Sturge and J. Taylor. 2008. *Freebase: a collaboratively created graph database for structuring human knowledge.* In proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 1247-1250.

A. Budanitsky. 1999. *Lexical semantic relatedness* *and its application in natural language processing.* Technical Report CSRG-390, Computer Systems Research Group, University of Toronto.

P. Buitelaar, D. Olejnik and M. Sintek. 2004. *A Protégé Plug-in for Ontology Extraction from Text Based on Linguistic Analysis.* In proceedings of the 1st European Semantic Web Symposium, pp. 31-44.

M. Ciaramita, A. Gangemi, E. Ratsch, J. Saric and I. Rojas. 2005. *Unsupervised Learning of Semantic Relations Between Concepts of a Molecular Biology Ontology.* In proceedings of the 19th International Joint Conference on Artificial Intelligence, pp. 659-664.

J. Edmonds. 1967. *Optimum branchings.* Journal of Research of the National Bureau of Standards, 71, pp. 233-240.

S. Fodeh, B. Punch and P. N. Tan. 2011. *On Ontology-driven Document Clustering Using Core Semantic Features.* Knowledge and information systems, 28(2), pp. 395-421.

H. N. Fotzo and P. Gallinari. 2004. *Learning "Generalization/Specialization" Relations between Concepts - Application for Automatically Building Thematic Document Hierarchies.* In proceedings of the 7th International Conference on Computer-Assisted Information Retrieval.

M. Geffet and I. Dagan. 2005. *The Distributional Inclusion Hypotheses and Lexical Entailment.* In proceedings of the 43rd Annual Meeting of the ACL, pp. 107-114.

R. Girju, A. Badulescu, and D. Moldovan. 2003. *Learning Semantic Constraints for the Automatic Discovery of Part-Whole Relations.* In proceedings of the NAACL, pp. 1-8.

S. M. Harabagiu, S. J. Maiorano and M. A. Pasca. 2003. *Open-Domain Textual Question Answering Techniques.* Natural Language Engineering, 9(3): pp. 1-38.

M. A. Hearst. 1992. *Automatic Acquisition of Hyponyms from Large Text Corpora.* In proceedings of the 14th Conference on Computational Linguistics, pp. 539-545.

D. Klein and C. D. Manning. 2003. *Accurate Unlexicalized Parsing.* In proceedings of the 41st Annual Meeting of the ACL, pp. 423-430.

Z. Kozareva, E. Riloff, and E. H. Hovy. 2008. *Semantic Class Learning from the Web with Hyponym Pattern Linkage Graphs.* In proceedings of the 46th Annual Meeting of the ACL, pp. 1048-1056.

Z. Kozareva and E. Hovy. 2010. *A Semi-supervised Method to Learn and Construct Taxonomies Using the Web.* In proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1110-1118.

C. Matuszek, J. Cabral, M. J. Witbrock and J. DeOliveira. 2006. *An Introduction to the Syntax and Content of Cyc.* In proceedings of the AAAI Spring Symposium: Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering, pp. 44-49.

G. A. Miller. 1995. *WordNet: a Lexical Database for English.* Communications of the ACM, 38(11), pp. 39-41.

R. Navigli and P. Velardi, 2004. *Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites.* Computational Linguistics, 30(2), pp. 151-179.

R. Navigli, P. Velardi and S. Faralli. 2011. *A Graph-based Algorithm for Inducing Lexical Taxonomies from Scratch.* In proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 1872-1877.

R. Snow, D. Jurafsky and A. Y. Ng. 2006. *Semantic Taxonomy Induction from Heterogenous Evidence.* In proceedings of the 21st International Conference on Computational Linguistics, pp. 801-808.

P. Velardi, S. Faralli and R. Navigli. 2012. *Ontolearn Reloaded: A Graph-based Algorithm for Taxonomy Induction.* Computational Linguistics, 39(3), pp. 665-707.

W. Wentao, L. Hongsong, W. Haixun, and Q. Zhu. 2012. *Probase: A probabilistic taxonomy for text understanding.* In proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 481-492.

D. Widdows and B. Dorow. 2002. *A Graph Model for Unsupervised Lexical Acquisition.* In proceedings of the 19th International Conference on Computational Linguistics, pp. 1-7.

W. Wong, W. Liu and M. Bennamoun. 2007. *Tree-traversing ant algorithm for term clustering based on featureless similarities.* Data Mining and Knowledge Discovery, 15(3), pp. 349-381.