Large-scale Dependency Knowledge Acquisition and its Extrinsic Evaluation Through Word Sense Disambiguation

Ping Chen [*] Wei Ding [†] Chris Bowes [*] David Brown [*]

Abstract

Knowledge plays a central role in intelligent systems. Manual knowledge acquisition is very inefficient and expensive. In this paper, we present (1) an automatic method to acquire a large amount of lexical-dependency knowledge, and (2) an innovative knowledge representation model to effectively minimize the impact of noise and improve knowledge quality. We also propose a new type of knowledge base evaluation – extrinsic evaluation, which evaluates knowledge by its impact to an external application. In our experiments we adopt Word Sense Disambiguation (WSD) as the extrinsic evaluation measure. Due to the lack of sufficient knowledge, existing WSD methods either are brittle and only capable of processing a limited number of topics or words, or provide only mediocre performance in real-world settings. With the support of acquired knowledge, our unsupervised WSD system significantly outperformed the best unsupervised systems participating in SemEval 2007, and achieved the disambiguation accuracy approaching top-performing supervised systems.

1 Introduction

Knowledge is critical in building intelligent systems. Without sufficient knowledge, computers often exhibit brittle behaviors, and can only carry out tasks that have been fully foreseen by their designers [5]. However, many real-world applications require a large amount of high-quality knowledge, which results in a severe knowledge acquisition bottleneck. Recently large-scale knowledge acquisition has attracted a lot of interest in Artificial Intelligence and Computational Linguistics.

While much work has been done on general-purpose knowledge base construction, the goal of our work is to acquire and evaluate a large amount of lexical-dependency knowledge. Lexical-dependency knowledge consists of dependency relations generated by dependency parsing of text [6]. A dependency relation is an asymmetric binary relation between two words, one called head or governor, and the other called dependent or modifier [9]. In dependency grammars a sentence is represented as a set of dependency relations, which normally form a tree that connects all the words in the sentence. For example, “blue sky” contains one dependency relation: “sky → blue”, where “sky” is the head, and “blue” is the dependent. Lexical-dependency knowledge can be used in many lexicon-level Natural Language Processing applications. For example, parsing “Colorless green ideas sleep furiously” will generate the following dependency relations, “idea → green”, “idea → colorless”, “sleep → idea”, and “sleep → furiously”, and we can conclude that the sentence makes totally no sense since none of its dependency relations are semantically valid. Similarly dependency knowledge can also be used to catch spelling errors by checking semantic dependency violations. In the rest of this paper, related work is discussed in Section 2. Section 3 describes how to acquire and represent lexical-dependency knowledge, which is evaluated with Word Sense Disambiguation in section 4, and the experiment results are presented in section 4.2. We conclude in section 5.

2 Related Work

Knowledge acquisition can be manual or automated based on the source of knowledge. With careful handcrafting manual approaches can generate high quality knowledge, but only in a small scale due to high cost. Automated approaches are more efficient in large-scale acquisition, but trustworthy of acquired knowledge is often questioned since many methods can not provide any quality assessment results.

[*] Dept. of Computer and Math. Sciences, University of Houston-Downtown, Email: chenp@uhd.edu

[†] Department of Computer Science, University of Massachusetts-Boston, Email: ding@cs.umb.edu

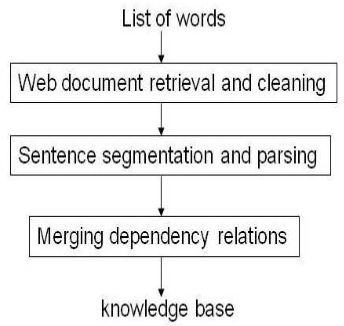

List of words
Web document retrieval and cleaning
Sentence segmentation and parsing
Merging dependency relations
knowledge base

Figure 1. Knowledge Acquisition Process

Manual large-scale knowledge acquisition systems include WordNet [3], Cyc Project [8], ConceptNet [14]. Manual approaches are labor-intensive. Even with long-time efforts (both WordNet and Cyc started about 20 years ago) of many human beings (over 12,000 people contributed to ConceptNet), building a comprehensive knowledge base is still remote.

Automated acquisition methods include MindNet [12], KnowItAll [2], ASKNet [4], etc. One major concern with an automatically-built knowledge base is its quality. Evaluation of large knowledge bases is still an open question. Currently many knowledge bases are often only assessed by statistical measures or small-scale manual evaluation [13].

Although it is generally agreed that knowledge plays a central role in intelligent information systems, current knowledge bases often can not provide sufficient support for real-world applications. In our work, instead of general knowledge acquisition, we aim to collect specific knowledge at a large scale to support a real-world application, Word Sense Disambiguation.

3 Dependency Knowledge Acquisition

Figure 1 shows an overview of our knowledge acquisition and representation system, and here are details about each step.

3.1 Web Document Retrieval and Cleaning

The goal of this step is to collect as much as possible valid text, and preferably the collection is also diverse and contain many different words. We found that Web documents are more suitable than static text collections for this purpose. Billions of documents exist in the World Wide Web, and millions of Web pages are created and updated everyday. Such a huge dynamic text collection is an ideal source to provide broad and up-to-date knowledge. The major concern about Web documents is inconsistency of their quality, and many Web pages are spam or contain erroneous information. However, factual errors in Web pages will not hurt the quality of lexical dependency knowledge. Instead, the quality of this kind of knowledge is affected by broken sentences of poor linguistic quality and invalid word usage, e.g., sentences like "Colorless green ideas sleep furiously" that violate commonsense knowledge. To start the acquisition process, a word (noun, verb, adjective, and adverb) is submitted to a Web search engine as a query. Several search engines provide API's for research communities to retrieve Web pages automatically. In our experiments we used Google API's to retrieve up to 1,000 Web pages for each word. These Web pages are cleaned first, e.g., control characters and HTML tags are removed.

3.2 Sentence Segmentation and Parsing

Sentences in the Web pages are segmented simply based on punctuation (e.g., ?, !, .) and are sent to a dependency parser, which parses a sentence and generates a set of dependency relations in the format of "$word_1 \rightarrow word_2$", where $word_1$ is the head, and $word_2$ is the dependent. We adopt Minipar in the experiments. An evaluation with the SUSANNE corpus shows that Minipar achieves 89% precision with respect to dependency relations [6]. Neither our simple sentence segmentation approach nor Minipar parsing is 100% accurate, so a small number of invalid dependency relations may exist. However, their impact is minimized in the following merging step.

3.3 Merging Dependency Relations

After parsing, dependency relations are merged and saved in a knowledge base. The merging process is straightforward. Nodes from different dependency relations are merged into one node as long as they represent the same word. After merging dependency relations, we will obtain a new knowledge representation model – a weighted dependency graph with a word as a node, a dependency relation as an edge, and the number of occurrences of dependency relation as weight of an edge. This weight indicates the strength of semantic coherence of the head and the dependent. As a fully automatic knowledge acquisition process, it is inevitable to include erroneous dependency relations in the knowledge base. However, in a large text collection valid dependency relations tend to repeat far more times than invalid ones, so these erroneous edges will be assigned a relatively small weight and have minimal impact on the quality of acquired knowledge.

3.4 Extrinsic Knowledge Evaluation

Evaluation of a large knowledge base is not trivial due to its size, high complexity, and lack of standard. Currently either only some statistical measures (e.g., size) are calculated or a small piece of knowledge base is manually evaluated [13]. Manual evaluation is limited in size and subjective, and human evaluators do not always agree with each other. In this paper we propose and apply a new type of knowledge evaluation method – extrinsic evaluation. Extrinsic evaluation does not assess knowledge bases directly as with intrinsic evaluation methods. Instead, knowledge is used in applications, and its quality is assessed by the impact it brings to these applications. Utilization in multiple application will be ideal to avoid any bias towards any specific areas. The knowledge quality is generally positively associated with the application's performance. If a knowledge base provides mostly noisy or erroneous knowledge, the performance will be negatively impacted. In next section we will discuss how to evaluate our knowledge base through its application in Word Sense Disambiguation.

4 Extrinsic Evaluation Through WSD

In many natural languages, a word can represent multiple meanings/senses. Word Sense Disambiguation (WSD) is the process of determining which sense of a polysemous word is used in a given context. WSD is a long-standing problem in Computational Linguistics, and has important application in many Natural Language Processing tasks. WSD methods use the context of a word to disambiguate its senses, and the context information can come from either sense-annotated text or other knowledge resources. Usually WSD techniques can be divided into 3 categories [1], dictionary-based methods, supervised methods [11], unsupervised methods[7]. Disambiguation of a limited number of words is not hard, and necessary context information can be carefully collected and hand-crafted to achieve high disambiguation accuracy. However, such approaches suffer a significant performance drop in practice when vocabulary is not limited.

4.1 Our Unsupervised WSD Method

WSD is often an unconscious process to human beings. With a dictionary and sample sentences/phrases an average educated person can correctly disambiguate most polysemous words. Inspired by human WSD process, we choose an electronic dictionary and unannotated text samples as knowledge source for our WSD system. Both dictionary and text samples can be automatically accessed, provide an excellent coverage of word meanings and usage, and are actively updated to reflect the current state of natural languages.

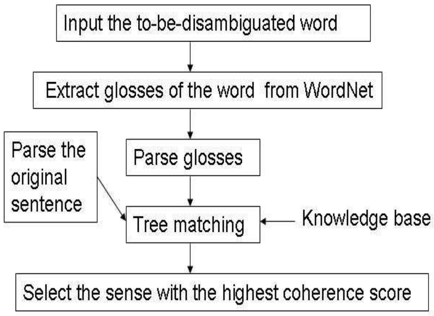

Input the to-be-disambiguated word
Extract glosses of the word from WordNet
Parse the original sentence
Parse glosses
Tree matching
Knowledge base
Select the sense with the highest coherence score

Figure 2. WSD Procedure

Based on this idea we set up our WSD procedure as shown in Figure 2. First both the original sentence that contains the to-be-disambiguated word and the glosses of to-be-disambiguated word are parsed. Then the parsing tree generated from each gloss is matched with the parsing tree of original sentence one by one. The gloss most semantically coherent with the original sentence will be chosen as the correct sense.

Figure 3 shows our WSD algorithm, and we illustrate our WSD algorithm through the following example. Assume we try to disambiguate "company" in the sentence "A large software company hires many computer programmers". "company" has 9 senses as a noun in WordNet 2.1. Let's pick the following two glosses to go through the WSD process.

- an institution created to conduct business
- small military unit

First we parse the original sentence and two glosses, and get three weighted parsing trees as shown in Figure 4. All weights are assigned to nodes/words in these parsing trees. In the parsing tree of the original sentence the weight of a node is reciprocal of the distance between this node and to-be-disambiguated node - "company" (line 12 in Figure 3). In the parsing tree of a gloss the weight of a node is reciprocal of the level of this node in the parsing tree (line 16 in Figure 3). Assume that the lexical-dependency knowledge base contains the dependency relations shown in Figure 5.

Each dependency edge is assigned a weight ranging in [0, 1] based on its frequency in the acquisition process. A frequent edge will be assigned a high value (close to 1), otherwise it has a low value (close to 0). Now we load the dependent words of each word in gloss 1 from the knowledge base (line 14, 15 in Figure 3), and

```
**Input:** Glosses from WordNet;
$S$: the sentence to be disambiguated;
$G$: the knowledge base generated in Section 3;
1. Input a sentence $S$, $W = \{w|\ w$'s part of speech
       is noun, verb, adjective, or adverb, $w \in S\}$;
2. Parse $S$ with a dependency parser, generate
       parsing tree $T_S$;
3. For each $w \in W$ {
4.     Input all $w$'s glosses from WordNet;
5.     For each gloss $w_i$ {
6.         Parse $w_i$, get a parsing tree $T_{wi}$;
7.         score = TreeMatching($T_S, T_{wi}$);
8.     }
9.     Choose the sense with the highest score as
           the correct sense;
10. }

**TreeMatching($T_S$, $T_{wi}$)**
11. For each node $n_{Si} \in T_S$ {
12.     Assign weight $w_{Si} = \frac{1}{l_{Si}}$, $l_{Si}$ is the
        length between $n_{Si}$ and $w_i$ in $T_S$;
13. }
14. For each node $n_{wi} \in T_{wi}$ {
15.     Load its dependent words $D_{wi}$ from $G$;
16.     Assign weight $w_{wi} = \frac{1}{l_{wi}}$, $l_{wi}$ is the
        level number of $n_{wi}$ in $T_{wi}$;
17.     For each $n_{Sj}$ {
18.         If $n_{Sj} \in D_{wi}$
19.            calculate connection strength $s_{ji}$
               between $n_{Sj}$ and $n_{wi}$;
20.            score = score + $w_{Si} \times w_{wi} \times s_{ji}$;}}
21. Return score;
```

Figure 3. WSD Algorithm

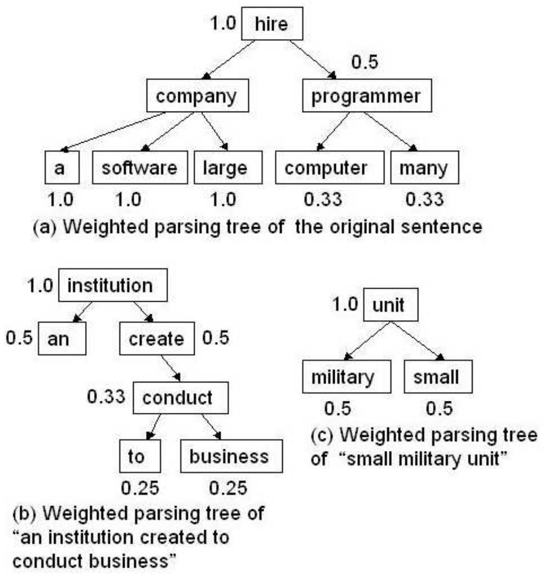
we get {large} for “institution” and {large, software} for “business”. In the dependent words of “company”, “large” belongs to the dependent word sets of “institution” and “business”, and “software” belongs to the dependent word set of “business”, so the coherence score of gloss 1 is calculated as (line 19, 20 in Figure 3):

1.0×1.0×0.7+1.0×0.25×0.8+1.0×0.25×0.9=1.125

We go through the same process with the second gloss “small military unit”. “Large” is the only dependent word of “company” appearing in the dependent word set of “unit” in gloss 2, so the coherence score of gloss 2 in the current context is:

$1.0 \times 1.0 \times 0.8 = 0.8$

After comparing the coherence scores of two glosses, we choose sense 1 of “company” as the correct sense (line 9 in Figure 3).



(a) Weighted parsing tree of the original sentence

(b) Weighted parsing tree of “an institution created to conduct business”

(c) Weighted parsing tree of “small military unit”

Figure 4. Weighted parsing trees of the original sentence and two glosses of “company”

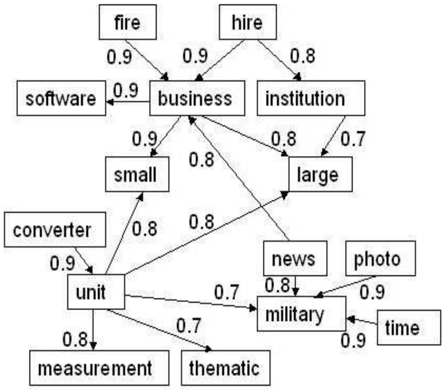


Figure 5. A fragment of lexical-dependency knowledge base

4.2 Experiment

We have evaluated our method using SemEval-2007 Task 07 (Coarse-grained English All-words Task) test set [10]. The task organizers provide a coarse-grained sense inventory created with SSI algorithm, training data, and test data. We followed the knowledge acquisition and WSD process described in Section 3.

The disambiguation results are shown in Table 1, and our system “TreeMatch” is marked in bold. For comparison we also listed the results of the top 2 systems and top 2 unsupervised systems participating in SemEval-2007 Task 07. Both of the top systems (UoR-

System	Attempted	Precision	Recall	F1
UoR-SSI	100.0	83.21	83.21	83.21
NUS-PT	100.0	82.50	82.50	82.50
TreeMatch	**100.0**	**73.60**	**73.60**	**73.60**
SUSSZ-FR	72.8	71.73	52.23	60.44
SUSSX-C-WD	72.8	54.54	39.71	45.96

Table 1. Overall disambiguation scores (Our system "TreeMatch" is marked in bold)

SSI and NUS-PT) are supervised systems, which used annotated resources (e.g., SemCor, Defense Science Organization Corpus) during the training phase. With the support of lexical-dependency knowledge base, our fully unsupervised WSD system significantly outperforms the top unsupervised systems (SUSSZ-FR and SUSSX-C-WD) and achieves performance approaching the top-performing supervised WSD systems, which clearly shows that our knowledge base contains broad-coverage and high-quality knowledge.

5 Conclusion

This paper presents an automatic lexical-dependency knowledge acquisition technique, and provides extrinsic evaluation through WSD. Using SemEval-2007 Task 7 WSD test set, our unsupervised WSD method achieved F-scores superior to existing unsupervised methods and approaching the top performing supervised systems. The experiment results clearly showed the effectiveness of our knowledge acquisition method and high quality of the collected knowledge.

Acknowledgments

This work is partially funded by NSF grants DUE 0737408, CNS 0851984, and UHD Scholar Academy.

References

[1] Agirre, E., Philip E. (eds.). 2006. Word Sense Disambiguation: Algorithms and Applications, Springer.

[2] Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, D., and Yates, A., Unsupervised named-entity extraction from the web: An experimental study. Artificial Intelligence, 165(1):91.134. 2005

[3] Fellbaum, C. WordNet: An Electronic Lexical Database, MIT press, 1998

[4] Harrington, B., Clark, S., ASKNet: Automated Semantic Knowledge Network, AAAI-07, Vancouver, Canada, 2007

[5] Lenat, D. B., Guha, R. V., Pittman, K., Pratt, D., and Shepherd, M., 1990. CYC: Towards programs with common sense. Communications of the ACM 33(8).

[6] Lin, D. 1998. Dependency-based evaluation of minipar. In *LREC Workshop on the Evaluation of Parsing Systems*, Granada, Spain.

[7] Lin, D., Using syntactic dependency as local context to resolve word sense ambiguity. ACL 1997.

[8] Matuszek, C., Cabral, J., Witbrock, M., DeOliveira, J., An Introduction to the Syntax and Content of Cyc. 2006 AAAI Spring Symposium, Arizona, 2006

[9] Melcuk, I. 1987. Dependency syntax: theory and practice. State University of New York Press.

[10] Navigli, R., Litkowski, K., Hargraves, O. 2007. Semeval-2007 task 07: Coarse-grained English all-words task. In *The 4th International Workshop on Semantic Evaluations*, Czech Republic.

[11] Novischi, A., Srikanth, M., Bennett, A. 2007. Lcc-wsd: System description for English coarse grained all words task at semeval 2007. In *The 4th International Workshop on Semantic Evaluations*, Czech Republic.

[12] Richardson, S.D., Dolan, W.B. and Vanderwende L., MindNet: Acquiring and Structuring Semantic Information from Text, ACL, 1998.

[13] Schubert, L. K., Tong, M. Extracting and evaluating general world knowledge from the Brown corpus. NAACL 2003 Workshop on Text Meaning.

[14] Singh, P., Liu, H., ConceptNet: a practical commonsense reasoning toolkit. BT Technology Journal, 22(4):211-226, 2004.