

Automatic Learning of Knowledge for Example-Based Disambiguation of Attachment

Naohiko URAMOTO

IBM Research, Tokyo Research Laboratory
1623-14 Shimo-tsuruma, Yamato-shi, Kanagawa-ken 242 Japan
uramoto@trl.ibm.co.jp

Abstract

This paper describes an attempt to improve the accuracy of example-based disambiguation with minimal human intervention. Two types of knowledge – interchangeable relationships and word-to-word dependencies with preference values – are learned automatically by using the enhanced bootstrapping method, and are stored in an acquired example base. Use of this example-base improved the accuracy of the disambiguation of attachment from 85.9% to 90.3%.

1 Introduction

The example-based approach [8] is now widely used in natural language processing applications such as machine translation and disambiguation [12, 13, 17]. However, few existing systems can cover a practical domain or handle a broad range of phenomena. Furthermore, existing systems pay no attention to multiple domains.

The most serious obstacle to realizing robust example-based systems is the coverage of example-bases. It is an open question how many examples are required for disambiguating sentences in a specific domain. Previously, it was believed to be easier to collect examples than to develop rules for resolving ambiguities. However, the coverage of each example is much more local than a rule, and therefore a huge number of examples for each domain is required in order to resolve realistic problems. Furthermore, even if an example-base becomes huge, *unknown words* cannot be handled, especially in domains such as that of computer manuals in which new commands and product names appear.

In order to overcome these problems, most systems employ the "bootstrapping" approach. In this framework, the systems function partly as tools for acquiring knowledge that they themselves can use. For example, in our example-based disambiguation system called Sentence Analyzer (SENA) [15, 16], an input of the system is an ambiguous dependency structure of a sentence. The ambiguities are resolved through the use of an example-base, and the disambiguated dependency structure is output. It is then checked by the user, and the correct structure is added to the example-base. New examples are acquired by iterating the process. Ideally, the more examples are provided, the less intervention is required from the user. However, in our experience, correcting and adding the examples are not easy tasks. If an elaborate process such as word-sense disambiguation is to be performed, elaborate examples are required. Furthermore, if the domain is changed, a new domain-dependent example-base must be built from scratch.

This paper describes an attempt to minimize the amount of human intervention required. Knowledge for the example-based disambiguation is acquired automatically by using the disambiguation system as a knowledge acquisition tool. Two types of knowledge

<i>Example-Base</i>	
<i>Domain</i>	<i>Size</i>
General	30,000
Computer manuals	27,000
Business letters	2,000

<i>Thesaurus</i>	
<i>Type</i>	<i>Size</i>
Synonyms	283,000 (11,006 entries)
Taxonyms	6,400

Table 1: Sizes of the Example-Bases and Thesaurus

are used. The first consists of relationships for calculating the similarity between words. Thesauri that include synonym and taxonym relationships are always used for this purpose in conventional systems. In this paper, *interchangeable* relationships are extracted from the text and the example-base. The second type of knowledge consists of word-to-word relationships. Dependencies, with preference values obtained from the output of SENA, are used to construct a "acquired example base."

2 Baseline System

An example-based disambiguation system called SENA [16, 15] is used as the baseline system. It resolves attachment and word-sense ambiguities by means of constraints and example-based preferences. This section briefly introduces the system and summarizes the results of the baseline experiments on disambiguation of attachments.

2.1 Example-Base and Thesaurus

The domain dependency of vocabulary has been accorded great importance in many studies; in the example-based approach, however, it has not received much attention.

In this paper, example-bases are classified into two categories: a single general example-base and various domain example-bases. The general example-base is always used, and one or more domain example-bases may be selected according to the context of the input sentences. The example-bases are assigned an order of priority. Basically, a domain example-base is given higher priority than the general example-base. An example from a high-priority is preferred.

Table 1 shows the current sizes of the example-bases. For the general example-base, the sentences were extracted from sample sentences and phrases in the Longman Dictionary of Contemporary English [10]. The biggest domain example-base is for computer manuals. It was created from definitions in the IBM Dictionary of Computing [5]. Synonym and is-a relationships were extracted from the New Collins Thesaurus [2] and Webster's Seventh New Collegiate Dictionary [1] (currently, no domain thesaurus is provided).

Each example-base is a set of head-modifier binary dependencies with semantic relations between words, such as (AGENT), (THEME), and ("in" GOAL). It was developed by a bootstrapping method with human correction.

2.2 Mechanism of Disambiguation

SENA resolves attachment and word-sense ambiguities with constraints and example-based preferences. The system uses the following knowledge:

- (a) Preferences from the example-base
- (b) Grammatical and semantic constraints
- (c) Statistical preferences
- (d) Heuristic rules

Each item of knowledge is applied to the ambiguities in the above order. If some preferences in (a) are calculated, and the most preferable attachment satisfies the constraints in (b), the disambiguation process is ended. If there are no preferences or there is a tie between preferences, (c) and (d) are applied, in that order.

((“store+V” *STORE-1) (“in” GOAL) (“disk” *DISK) 1)
 ((“store+V” *STORE-1) (“in” GOAL) (“storage-device” *DEVICE) 2)
 ((“store+V” *STORE-1) (“in” GOAL) (“cell” *CELL) 1)
 ((“store+V” *STORE-1) (“in” GOAL) (“computer” *COMPUTER-1) 4)
 ((“store+V” *STORE-1) (“in” GOAL) (“storage” *STORAGE-2) 3)
 ((“store+V” *STORE-1) (“in” GOAL) (“format” *FORMAT-1) 1)
 ((“store+V” *STORE-1) (“in” GOAL) (“data-network” *NETWORK-3) 1)

Figure 1: Examples for R1

((“program+N” *PROGRAM-1) (“in” TIME-SPACE-RANGE) (“profile+N” *PROFILE) 5)
 ((“program+N” *PROGRAM-1) (“in” TIME-SPACE-RANGE) (“data-storage+N” *STORAGE-3) 1)
 ((“program+N” *PROGRAM-1) (“in” TIME-SPACE-RANGE) (“publication+N” *PUBLISH-1) 1)
 ((“program+N” *PROGRAM-1) (“in” TIME-SPACE-RANGE) (“form+N” *FORM-1) 2)
 ((“program+N” *PROGRAM-2) (“in” TIME-SPACE-RANGE) (“group+N” *GROUP-1) 1)

Figure 2: Examples for R2

Semantic constraints (selectional constraints and rules for subcategorization) and grammatical constraints (for example, no crossing of read-modifier relationships) are classified under (b). Statistical preferences (c) are calculated for relationships between a preposition and the words that are candidates for attachment. This is a variation of Hincle and Rooth's approach [4]. The catch-all rules (d) are a small set of heuristic rules, such as a, rule whereby innermost attachment is preferred.

Let us examine the process for resolving ambiguity in the attachment of prepositional phrases. In the sentence

(S1) **The system can store a new program in the repository,**
 there are two candidates for the attachment of the prepositional phrase “in the repository.” They are represented by the following head-modifier relationships:

(R1) (“store+V” (“in”) “repository+N”)
 (R2) (“program + N” (“in”) “repository+N”)

In R1 the noun “repository” modifies the verb “store” with “in,” while in R2, it modifies the noun “program.” First, SENA searches for examples whose heads match those of the candidates.

Figures 1 and 2 show the relevant examples from our example-bases for R1 and R2. They represent the head-modifier relationships, including word-senses, a relation label between the word-senses, (e.g. “in”), and a frequency. If a relationship identical to either of the candidates R1 and R2 is found, a high similarity is attached to the candidate and the example (exact matching). Word-sense ambiguities are resolved by using the same framework [15]. In this case, each candidate represents a single word sense. For example, the word-sense *STORE-1 is preferred among the examples shown in Figure 1.

If no examples are obtained by the exact-matching process, the system executes the best-matching process, which is the most important mechanism in the example-based approach. Synonym or is-a relationships described in a thesaurus are used for the comparison. For example, if synonym relations are found between “repository” and “disk” in the first example for R1, a similarity whose value is smaller than that for exact matching is given to the examples. The most preferable candidate is selected by comparing all the examples in Figure 1 and computing the total similarity value for each candidate.

2.3 Baseline Experiment: Coverage and Accuracy

The system described above was used to conduct some experiments in attachment disambiguation. Five hundred sentences from a computer manual for personal computers were collected and divided into five test sets. None of the sentences was among those

	Set1	Set2	Set3	Set4	Set5	Total
Distinct words	840	755	538	477	885	923
One word only (%)	85.6	84.8	90.5	86.1	82.5	85.9
Head/modifier (%)	47.1	50.0	45.3	52.1	60.8	51.6

Figure 3: Coverage of the Examples

Type of knowledge	Accuracy (%)
(a) Examples	43.6
(b) Constraints	25.5
(c) Statistics	8.9
(d) Heuristics	7.2
(a)+(b)+(c)+(d)	85.2

Figure 4: Distribution of the Accuracy of Each Type of Knowledge

Type of knowledge	Accuracy (%)
(a) Examples	93.0
(b) Constraints	98.4
(c) Statistics	68.8
(d) Heuristics	68.4

Figure 5: Accuracy of Each Type of Knowledge

used to construct the original example-base. The attachment ambiguities in those test sentences, including pp-attachment and infinitive and relative pronoun clause attachment, were then resolved.

Figure 3 shows the coverage of the example-base for the test sentences. The second row (“One word only”) represents the coverage of the example-base for each word in the test set. Of the words in the test set, 85.9% appear in the example-base. The third row (“Head/modifier”) represents the coverage of the example-base for the head/modifier (binary) relationships in the test sets. The overall coverage (51.6%) is disappointing, since basically the accuracy of (pure) example-based disambiguation cannot improve the coverage.

Figure 4 shows the accuracy of the disambiguation. The overall accuracy for the disambiguation of attachment is 85.2%, of which pure example-based disambiguation contributes about half (43.6%). It has been claimed that it is difficult to resolve ambiguities solely by an example-based approach, in terms of accuracy and efficiency [15]. Figure 5 shows the accuracy of each type of knowledge. For example, in all the attachments determined by the pure example-based approach, 93.0% of the decisions were correct. It is natural that constraints should have the highest accuracy, and that the reliability of examples should also be high. The accuracy of statistics and heuristics is not so high, but they can work as catch-all rules.

3 Automatic Acquisition of Knowledge for Use in Disambiguation

As shown in the previous section, the pure example-based approach does not cover all ambiguities. One reason for this is the domain dependency of words and word-to-word relationships. For example, one of our domain example-bases was created for computer manuals about host computers and their applications. The test set used in the experiment, on the other hand, is from a manual for a personal computer. The word “diskette,” which is very common in the personal computer domain, does not appear in the example-base.

One of the advantages claimed for the example-based approach is the ease of collecting examples. However, the cost of this task is considerable, for the following two reasons. First, if elaborate processing is attempted, elaborate examples are required. Second, all the examples should be correct, and must therefore be checked by humans.

This section describes a method of acquiring knowledge automatically for example-based disambiguation in order to minimize human intervention. As mentioned in Section

(D1) (“remove+V” *REMOVE “diskette+N”) (THEME) (“diskette+N” *DISKETTE) 1.0)
 (D2) (“insert+V” *INSERT) (THEME) (“diskette+N” *DISKETTE) 2.0)
 (D3) (“install+V” *INSTALL) (“from” SOURCE) (“diskette+N” *DISKETTE) 0.75)
 (D4) (“diskette+N” *DISKETTE) (SUCH-THAT) (“instruct+V” *INSTRUCT) 2.04)
 (D5) (“diskette+N” *DISKETTE) (“from” SOURCE) (“drive+N” *DRIVE) 0.25)
 (D6) (“diskette+N” *DISKETTE) (“in” GOAL) (“drive+N” *DRIVE) 1.0)
 (D7) (“create+V” *CREATE-2) (THEME) (“diskette+N” *DISKETTE) 8.0)
 (D8) (“book+N” *ONLINE-BOOK) (“from” SOURCE) (“diskette+N” *DISKETTE) 0.24)

Figure 6: Dependencies of “diskette”

1. two types of knowledge are needed for example-based disambiguation: interchangeable relationships and word-to-word dependencies.

3.1 Estimating Interchangeable Relationships

If similarities between words are calculated, the coverage of examples can be expanded. Synonym or taxonym (is-a) relationships are used to calculate such similarities.

Synonym and taxonym relationships represent mainly semantic similarity. However, syntactic relationships are also useful for calculating similarity. In this paper, *interchangeable relationships* are used for this purpose. In a sense, such relationships include synonym and taxonym relationships. If a word in a sentence can be replaced by another word in some contexts, the words are interchangeable. Our claim is that if words are interchangeable in sentences, they should have a strong similarity. This section describes two methods for estimating interchangeability by using word-to-word dependencies.

Conjunctive relationships, which are common in technical documents, provide a good clue to the interchangeability. The interchangeable relationships extracted from the conjunctive structures in test set 1, which we used in Section 2, are as follows:

installation/configuration, server/requester, hardware/software, user/group, ID/password

Other interchangeable relationships can be learned by calculating the similarities between dependencies. They are used to find the words that can be assumed to be interchangeable with unknown words.

In the sentence

(S2) **Insert the correct printer driver diskette in the diskette drive,**

the word “diskette” does not appear in the example-base or thesaurus. The existence of unknown words is inevitable when one is dealing with the disambiguation of sentences in practical and multiple domains. Computer manuals, for example, contain many special terms such as names of commands and products, but there are no thesauri for such highly domain-specific words.

However, an unknown word often appears many times in the same text [9]. By comparing the dependencies that include the unknown word and examples in the example-base, words that are interchangeable with the unknown word can be acquired. Figure 6 shows the dependencies of “diskette” from test sets 1-5. The last numbers in the right column are the preference values calculated by SENA. In Figure 6, D2 and D5 are from the sentence S2. The dependencies include ambiguities. Some systems, including one based on our early work [16], reduce the reliability of ambiguous dependencies (simplicistically, the reliability for D2 and D5 is 0.5, and that for the other dependencies is 1.0). In this paper, the preference values, which are the results of applying knowledge, are used.

Words that have the same dependencies are searched for in the example-base. For instance, from D1, the following dependencies are extracted:

(D11) (“insert+V” *INSERT) (THEME) (“CD-ROM+N” *CD-ROM) 2)

IW	Preference Value	IW	Preference Value
"file"	40.0	"data"	27.0
"view"	40.0	"library"	24.4
"job"	35.0	"definition"	24.0
"id"	32.0	"service"	24.0
"table"	28.0	"user"	24

Figure 7: Words Judged without Filtering to Be Interchangeable with "diskette"

IW	Preference Value	IW	Preference Value
"book"	6.75	"row"	4.0
"drive"	6.0	"CD-ROM"	3.5
"information"	5.0	"adapter"	3.5
"server"	5.0	"character"	3.0
"disk"	4.25	"data"	3.0

Figure 8: Words Judged with Filtering to Be Interchangeable with "diskette"

(D12) ((“insert+V” *INSERT) (THEME) (“data+N” *DATA) 1)

(D13) ((“insert+V” *INSERT) (THEME) (“disk+N” *DISK) 1)

The words “CD-ROM,” “data,” and “disk” are candidates for interchangeability with “diskette,” and are stored together with their preference values. The process is applied to D2-D8, and Figure 7 shows the top ten interchangeable words (IWs).

Note that the words do not seem to be similar to “diskette.” The reason for this is the system’s ignorance of the *typicalness* of dependencies. For example, from D7, the word that functions as the theme of the verb “create” is selected. Since the word appears many times in the example-base, the candidate from D7 is weighted. However, a wide range of words can function as the theme of “create.” On the other hand, the verb “insert” restricts the class of words. To eliminate this effect, a filtering process is invoked when dependencies such as D1-D7 are acquired. If a dependency has too many examples in the example-base, it is not selected. The threshold is set at 100 in this paper.

Figure 8 shows the results of using the filtering to acquire interchangeable words. By eliminating dependencies whose coverage is too wide, appropriate interchangeable words can be acquired. These results (in which “diskette” is similar to “disk”) and the above D13 make it possible to resolve the ambiguity in the S2.

3.2 Acquiring of Word-to-Word Relationships

The output of SENA consists of word-to-word dependencies with preference values. Disambiguation of sentence S1

(S1) **The system can store a new program in the repository**

led to the output of

(E1) ((“store+V” *STORE-1) (THEME) (“program+N” *PROGRAM-1) 1)

(E2) ((“store+V” *STORE-1) (“in” GOAL) (“repository” *REPOSITORY) 1.2)

(E3) ((“program+N” *PROGRAM-1) (“in” TIME-SPACE-PLANE) (“repository” *REPOSITORY) 0.4)

(E4) ((“system+N” *SYSTEM) (AGENT) (“store” *STORE-1) 1)

The final number in each relationship is the preference calculated by SENA. In the human-aided approach, these relationships are checked by humans, and only correct relationships, with appropriate preferences, are added to the example-base. In the automatic

learning approach, the preference value is used instead of the frequency. In the learning process, the *acquired examples* are created by using these dependencies. The acquired examples are given a lower priority than example-bases that have been checked by humans. Therefore, the acquired examples does not conflict with "true" examples from the human-checked example-bases.

4 Experiment

To evaluate the automatically acquired knowledge, an experiment in attachment disambiguation was performed, using the test sets described in Section 2. The order of priority of the example-bases is (1) general example-base, (2) domain example-base (for computer manuals), and (3) acquired examples. Interchangeable relationships are added to the general thesaurus. In the experiment, use of acquired knowledge improved the accuracy of the disambiguation of attachment from 85.9% to 90.3%. One of the causes of failure was an insufficient number of examples, which prevented our system from estimating the interchangeable relationships of words in the test set. If frequent examples in the text to be disambiguated do not appear in the example-base, a wrong estimate is acquired. Minimum intervention is required to resolve this phenomenon. Human intervention can be reduced by preferring examples that allow disambiguation of multiple attachments.

5 Related Work

The basic idea of the disambiguation of prepositional attachment was described in a paper on Nagao's Dependency Analyzer [7], which was a predecessor of SENA. The system uses both grammatical constraints and example-based preference. Sumita et al. resolved ambiguities in pp-attachment by using only the example-based approach in a restricted domain [13]. In their approach, the problems of data sparseness and domain-dependency are not considered. Kinoshita et al. proposed a disambiguation method that uses dependencies in the text to be processed [6]. Their method uses only dependencies with frequencies, while ours estimates both interchangeable relationships and dependencies with preferences.

Various methods for corpus-based disambiguation of prepositional phrase attachment have been proposed [4, 11]. Hindle et al. used statistics on the co-occurrence of a main verb and a preposition, and of the head of noun phrase and a preposition (in the case of S2, the co-occurrence of insert+into and diskette+into are compared). This method is also used in our system, and our experiment confirmed its usefulness. However, many training data are needed to improve its accuracy. Furthermore, knowledge acquired by the statistical method cannot reflect human updates. The advantage of example-based disambiguation is that the user can manipulate the example-base directly. Brill et al. proposed a rule-based approach in which partially disambiguated (annotated) text [3] is used. The learned rules are powerful, but over-generalization is inevitable. Our acquired examples do not conflict with the existing example-bases, since a priority is attached to each example-base.

6 Conclusion

This paper has presented methods for minimizing human intervention and for maximizing improvement in the disambiguation of attachment. Estimating the interchangeability of words and word-to-word relationships improved the accuracy of the disambiguation from 85.9% to 90.3%.

The main advantage of the example-based approach is tractability of knowledge, since it provides a set of examples word-to-word relationships. However, it is not efficient to attempt to cover all ambiguities with examples. The appropriate level of knowledge for the disambiguation depends on the case. For example, it is difficult in the example-based framework to describe the preference that represents according to which prepositional

phrases related to time are attached to a verb phrase. On the other hand, there are many exceptions in natural language sentences that cannot be covered with rules. Seamless use of various types of knowledge is necessary. It is difficult to acquire sufficient knowledge solely by automatic learning, without any human guidance. A more efficient interactive learning process is also needed.

References

- [1] "Webster's Seventh New Collegiate Dictionary," G & C. Merriam, Springfield, Massachusetts, 1963.
- [2] "The New Collins Thesaurus," Collins Publishers, Glasgow, 1984.
- [3] E. Brill and P. Resnik. "A Rule-Based Approach to Prepositional Phrase Attachment Disambiguation," *Proceedings of COLING-94*, pp. 1198-1204, 1994.
- [4] D. Hindle and M. Rooth, "Structural Ambiguity and Lexical Relations," *Computational Linguistics*. Vol. 19, No. 1, pp. 103-120, 1993.
- [5] IBM Corporation. "IBM Dictionary of Computing," SC20-1699-07, 1988.
- [6] S. Kinoshita, M. Shimazu, and H. Hirakawa. "Better Translation with Knowledge Extracted from Source Text," *Proceedings of TMI-93*, pp. 240-252, 1993.
- [7] K. Nagao. "Dependency Analyzer: A Knowledge-Based Approach to Structural Disambiguation," *Proceedings of COLING-90*, pp. 282-287, 1990.
- [8] M. Nagao. "A Framework of a Mechanical Translation between Japanese and English by Analogy Principle," A. Elithorn and R. Banerji. (eds), "Artificial and Human Intelligence", NATO, 1984.
- [9] T. Nasukawa. "Discourse Constraint in Computer Manuals," *Proceedings of TMI-93*, pp. 183-194, 1993.
- [10] P. Procter. "Longman Dictionary of Contemporary English," Longman Group Limited, Harlow and London, England, 1978.
- [11] A. Raptanarkhi and S. Roukos. "A Maximum Entropy Model for Prepositional Phrase Attachment," *Proceedings of the ARPA Workshop on Human Language Technology*, pp. 242-247, 1994.
- [12] S. Sato and M. Nagao. "Towards Memory-Based Translation," *Proceedings of COLING-90*, pp. 146-152, 1990.
- [13] E. Sumita, O. Furuse, and H. Iida. "A Example-Based Disambiguation of Prepositional Phrase Attachment," *Proceedings of TMI 93*, pp. 80-91, 1990.
- [14] K. Takeda, N. Uramoto, T. Nasukawa, and T. Tsutsumi. "Shalt2 - A Symmetric Machine Translation System with Conceptual Transfer," *Proceedings of COLING-92*, pp. 1034-1038, 1992.
- [15] N. Uramoto. "Example-Based Word-Sense Disambiguation," *IEICE Transactions on Information and Systems*, Vol. E77-D, No. 2, 1994.
- [16] N. Uramoto. "A Best-Match Algorithm for Broad-Coverage Example-Based Disambiguation," *Proceedings of COLING-94*, pp. 717-721, 1994.
- [17] H. Watanabe. "A Similarity-Driven Transfer System," *Proceedings of COLING-92*, pp. 770-776, 1992.