A Best-Match Algorithm for Broad-Coverage Example-Based Disambiguation

Naohiko URAMOTO
IBM Research, Tokyo Research Laboratory
1623-14 Simotsuruma, Yamato-shi, Kanagawa-ken 242 Japan
uramoto@trl.vnet.ibm.com

Abstract

To improve the coverage of example-bases, two methods are introduced into the best-match algorithm. The first is for acquiring conjunctive relationships from corpora, as measures of word similarity that can be used in addition to thesauruses. The Second, used when a word does not appear in an example-base or a thesaurus, is for inferring links to words in the example-base by comparing the usage of the word in the text and that of words in the example-base.

1 Introduction

Improvement of coverage in practical domains is one of the most important issues in the area of example-based systems. The example-based approach [6] has become a common technique for natural language processing applications such as machine translation and disambiguation (e.g. [5, 10]). However, few existing systems can cover a practical domain or handle a broad range of phenomena.

The most serious obstacle to 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.

The Sentence Analyzer (SENA) was developed in order to resolve attachment, word-sense, and conjunctive ambiguities by using constraints and example-based preferences [11]. It has about 57,000 disambiguated head-modifier relationships and about 300,000 synonyms and is-a binary-relationships. Even so, lack of examples (no relevant examples) accounted for 46.1% of failures in a experiment with SENA [12].

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 is required in order to resolve realistic problems. There has been some corpus-based research on how to acquire large-scale knowledge automatically in order to cover the domain to be disambiguated, but there are still major problems to be overcome.

First, semantic knowledge such as word-sense cannot be extracted by automatic corpus-based knowledge acquisition. The example-base in SENA is developed by using a bootstrapping method. However, the results of word-sense disambiguation must be checked by a human, and word-senses are tagged to only about a half of all the examples, since the task is very time-consuming.

A second difficulty in the example-based approach is the algorithm itself, namely, the *best-match* algorithm, which was used in earlier systems built around a thesaurus that consisted of a hierarchy of is-a or synonym relationships between words (word-senses).

This paper proposes two methods for improving the coverage of example-bases. The selected domain is that of sentences in computer manuals. First, knowledge that represents a type of similarity other than synonym or is-a relationships is acquired. As one measurement of the similarity, interchangeability between words can be used. In this paper, two types of the relationship reflect such interchangeability. First, the elements of coordinated structures are good clues to the interchangeability of words. Words can be extracted easily from a domain-specific corpus, and therefore the example-base can be adapted to the specific domain by using the domain-specific relationships.

If there are no examples and relations in the thesaurus, the example-base gives no information for disambiguation. However, the text to be disambiguated provides useful knowledge for this purpose [7, 3]. The relationships between words in the example-base and an unknown word can be guessed by comparing that word's usage in extracted examples and in the text.

2 A Best-Match Algorithm

In this section, conventional algorithms for example-based disambiguation, and their associated problems, are briefly introduced. The algorithms of most example-based systems consist of the following three steps[1]:

[1] In some systems, the exact-match and the best-match are merged.

("store+V" *store1 "in" "disk" *disk 1)
("store+V" *store1 "in" "storage-device" *device 2)
("store+V" *store1 "in" "cell" *cell 1)
("store+V" *store1 "in" "computer" *computer1 4)
("store+V" *store1 "in" "storage" *storage2 3)
("store+V" *store1 "in" "format" *format1 1)
("store+V" *store1 "in" "data-network" *network3 1)

Fig. 1: Examples for R1

("program+N" *prog1 "in" "profile+N" *profile 5)
("program+N" *prog1 "in" "data-storage+N" *storage3 1)
("program+N" *prog1 "in" "publication+N" *publication1 2)
("program+N" *prog1 "in" "form+N" *form1 2)
("program+N" *prog2 "in" "group+N" *group1 1)

Fig. 2: Examples for R2

1. Searching for examples
2. Exact matching
3. Best matching with a thesaurus

Suppose the prepositional phase attachment ambiguity in S1 is resolved by using these steps.

(S1) A managed AS/400 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" (PP "in") "repository+N")
(R2) ("program+N" (PP "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 the candidate. Figures 1 and 2 show the relevant examples 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 [12]. In this case, each candidate represent each word sense. For example, the word-sense *store1 is preferred among the examples shown in Fig. 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. For the comparison, synonym or is-a relationships described in a thesaurus are used. For example, if synonym relations are found between "repository" and "disk" in the first example for the 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 examples in Fig. 1 and computing the total similarity value for each candidate. If multiple candidates have the same similarity values, the frequency of the example and some heuristics (for example, innermost attachment is preferred) are used to weight the similarities.

Experience with SENA reveals two problems that prevent an improvement in the performance of the best-matching algorithm. First, the approach is strongly dependent on the thesaurus. Many systems calculate the similarity or preference mainly or entirely by using the hierarchy of the thesaurus. However, these relationships indicate only a certain kind of similarity between words. To improve the coverage of the example-base, other additional types of knowledge are required, as will be discussed in the following sections.

Another problem is the existence of *unknown words*; that is, words that are described in the system dictionary but do not appear in the example-base or the thesaurus. In SENA, the New Collins Thesaurus [1] is used to disambiguate sentences in computer manuals. Many unknown words appear, especially nouns, since the thesaurus is for the general domain. Therefore, a mechanism for handling the unknown words is required. This is covered in Chapter 4.

3 Knowledge Acquisition for Robust Best-Matching

As described in the previous section, the best-matching algorithm is a basic element of example-based disambiguation, but is strongly dependent on the thesaurus. Nirenburg [8] discusses the type of knowledge needed for the matching; in his method, morphological information and antonyms are used in addition to synonym and is-a relationships. This section discusses the acquisition of knowledge from other aspects for a broad-coverage best-match algorithm.

3.1 Acquisition of Conjunctive Relationships from Corpora

The New Collins Thesaurus, which is used in SENA as a source of synonym or is-a relationships, gives the following synonyms of "store":

store:
accumulate, deposit, garner, hoard, keep, etc.

In our example-base, there are few examples for any of the words except "keep," since the example-base was developed mainly to resolve sentences in technical documents such as computer manuals. When the domain is changed, the vocabulary and the usage of words also change. Even a general-domain thesaurus sometimes does not suit a specific domain. Moreover, development of a domain-specific thesaurus is a time-consuming task.

The use of synonym or is-a relationships suggests the hypothesis that from the viewpoint of the example-based approach, a word in a sentence can be replaced by its synonyms or taxonyms. That is, it supports the existence of the (virtual) example S1' when "store" and "keep" have a synonym relationship.

(S1') A managed AS/400 system can *keep* a new program in the repository.

Interchangeability is an important condition for calculating similarity or preferences between words. Our claim is that if words are interchangeable in sentences, they should have strong similarity.

In this paper, conjunctive relationships, which are common in technical documents, are proposed as relationships that satisfy the condition of interchangeability. Sentences in which the word "store" is used as an element of coordinated structure can be extracted from computer manuals, as following examples show:

(1) The service retrieves, formats, and stores a message for the user.
(2) Delete the identifier being stored or modified from the table.
(3) This EXEC verifies and stores the language defaults in your file.
(4) You use the function to add, store, retrieve, and update information about documents.

From the sentences, the following words that are interchangeable with "store" are acquired:

store: *retrieve, format, modify, verify, add, update*

Often the words share case-patterns, which is a useful characteristic for determining interchangeability. Another reason we use conjunctive relationships is that they can be extracted semi-automatically from untagged or tagged corpora by using a simple pattern-matching method. We extracted about 700 conjunctive relationships from untagged computer manuals by pattern matching. The relationships include various types of knowledge, such as 1(a) antonyms (e.g. "private" and "public"), (b) sequences of actions (e.g. "load" and "edit"), (c) (weak) synonyms (e.g. "program" and "service"), and (d) part-of relationships (e.g. "tape" and "device"). Another merit of conjunctive relationships is that they reflect domain-specific relations.

3.2 Acquisition from Text to Be Disambiguated

If there are no examples of a word to be disambiguated, and the word does not appear in the thesaurus, no relationships are acquired.

The existence of words that are unknown to the example-base and the thesaurus is inevitable when one is dealing with the disambiguation of sentences in practical domains. Computer manuals, for example, contain many special nouns such as names of commands and products, but, there are no thesauruses for such highly domain-specific words.

One way of resolving the problem is to use the text to be processed as the most domain-specific example-base. This idea is supported by the fact that most word-to-word dependencies including the unknown words appear many times in the same text. Nasukawa [7] developed the Discourse Analyzer (DIANA), which resolves ambiguities in a text by dynamically referring to contextual information. Kinoshita et al. [3] also proposed a method for machine translation by parsing a complete text in advance and using it as an example-base. However, neither system works for unknown words, since both use only dependencies that appear explicitly in the text.

4 An Algorithm to Search for Unknown Words

We first give an enhanced best-match algorithm for disambiguation. The steps given in Chapter 2 are modified as follows:

1. Searching for examples
2. Exact matching
3. Best matching with a thesaurus and conjunctive relationships
4. Unknown-word-matching using a context-base

The outline of the the algorithm is as follows: Sentences in the text to be processed are parsed in advance, and the parse trees are stored as a *context-base.* The context-base can include ambiguous word-to-word dependencies, since no disambiguation process is executed. Using an example-base and the context-base, the sentences in the text are disambiguated sequentially. If an ambiguous word does not appear in an example-base or in the thesaurus, an unknown word search is executed (otherwise, the conventional best-match process is executed.) The unknown-word-matching process includes the following steps:

1. The dependencies that include the unknown word are extracted from the context-base.
2. A candidate set of words that is interchangeable with the unknown word is searched for in the example-base by using the context dependency.
3. The candidate set acquired in step 2 is compared with the examples extracted for each candidate of interpretation. A preference value is calculated by using the sets, and the most preferred interpretation is selected.

Let us see how the algorithm resolves the attachment ambiguity in sentence S1 from Chapter 2, which is taken from a text (manual) for the AS/400 system.

(S1) A managed AS/400 system can store a new program in the repository.

The text that contains S1 is parsed in advance, and stored in the context-base. The results of the example search are shown in Fig. 1. There are two candidate relationships for the attachment of the prepositional phrase "in the repository".

(R1) ("store+V" (PP "in") "repository+N")
(R2) ("program+N" (PP "in") "repository+N")

The noun "repository" does not appear in the example-base or thesaurus, and therefore no information for the attachment is acquired.

Consequently, the word-to-word dependencies that contain "repository" are searched for in the context-base. The following sentences appear before or after S1 in the text:

(CB1) The repository can hold objects that are ready to be sent or that have been received from another user library.
(CB2) A distribution catalog entry exists for each object in the distribution repository.
(CB3) A data object can be loaded into the distribution repository from an AS/400 library.
(CB4) The object type of the object specified must match the information in the distribution repository.

From the sentences, the head-modifier relationships that contain the unknown word "repository" are listed. These relationships are called the *context dependency* for the word. The context dependency of "repository" is as follows:

(D1) ("hold+V" (subj) "repository+N"): 1
(D2) ("exist+V" (PP "in") "repository+N") : 0.5
(D3) ("object+N" (PP "in") "repository+N"): 0,5
(D4) ("load+V" (PP "into") "repository+N"): 1
(D5) ("information+N" (PP "in") "repository+N") : 0.5
(D6) ("match+V" (PP "into") "repository+N"): 0.5

The last number in each relation is the certainty factor (CF) of the relationship. The value is 1/(the number of candidates for the resolving ambiguity). For example, the attachment of "repository" in CB2 has two candidates, D2 and D3. Therefore, the certainty factors for D2 and D3 are 1/2.

For each dependency, candidate words (CB) in the context-base are searched for in the example-base. The words in the set can be considered as substitutable synonyms of the unknown word. For example, the WORDs that satisfy the relationship ("hold+V" (subj) WORD+N) in the case of D1 are searched for. The following are candidate words in the context-base for the word"repository."

CB1 = {I, user, cradle, rock} (for D1)
CB2 = {storage, transient data} (for D2)
CB3 = {condition, format, path, 1916, technique, control area} (for D3)
CB4 = {system38, facility} (for D4)
CB5 = {record} (for D5)
CB6 = {} (for D6)

The total set of candidate words (CB) of the "repository" is an union of CB1 through CB6. The set is compared with the extracted examples for each attachment candidate (Fig. 1). The words in the examples are candidate words in the example-base. By intersecting the candidate words in the context-base and the example-base, word that are interchangeable with the unknown word can be extracted. The intersections of each set are as follows:

For R1, CB∩C1 = {storage, format}
For R2, CB∩C2 = {}

This result means that "storage" and "format" have the same usage (or are interchangeable) in the text. The preference value P(R) for the candidate R with the interchangeable word w is calculated by the formula:

$$P(R) = \sum_{w}(CF) \times (frequency)$$

In this case, P(R1) = 0.5 × 1 + 0.5 × 1 = 1.0, and P(R2) = 0 (supposing that the frequency of the words is 1). As a result, R1 is preferred to R2.

If both sets of candidates are empty, the numbers of extracted examples are compared (this is called Heuristic-1). If there are no related words in this case, R1 is preferred to R2 (see Fig. 1). This heuristic indicates that "in" is preferred after "store," irrespective of the head word of the prepositional phrase.

5 Experimental Results

5.1 Example-Base and Thesaurus

An example-base for disambiguation of sentences in computer manuals is now being developed. Table 1 shows its current size. The sentences are extracted from examples in the Longman Dictionary of Contemporary English [9] and definitions in the IBM Dictionary of Computing [2]. Synonym and is-a relationships are extracted from the New Collins Thesaurus [1] and Webster's Seventh New Collegiate Dictionary [4].

Our example-base is a set of head-modifier binary dependencies with relations between word, such as (subject), (object), and (PP "in"). It was developed by a bootstrapping method with human correction. In SENA, the example-base is used to resolve three types of ambiguity: attachment, word-sense, and coordination. The level of knowledge depends on the type of ambiguity.

Table 1: Size of the Example-Base and Thesaurus

Example-Base	
Examples	57,170 binary relationships (in 9,500 sentences)
Distinct words	8,602
Thesaurus	
Synonyms	283,211 binary relationships (11,006 entries)
Is-a relations	6,353 binary relationships

Success with unknown word matching	52.4 (%)
Success with Heuristic-1	20.0 (%)
Failure	27.6 (%)

Fig. 3: Result of disambiguation

To resolve semantic ambiguities, the examples should be disambiguated semantically. On the other hand, structural dependencies can be extracted from raw or tagged corpora by using simple rules or patterns. In our approach, multilevel descriptions of examples are allowed: one example may provide both structural and word-sense information, while another may provide only structural dependencies. Word-senses are added to a half of the sentences in example-base.

5.2 Experiment

We did a small experiment on disambiguation of prepositional phrase attachment. First, we prepared 105 ambiguous test data randomly from 3,000 sentences in a computer manual. The format of the data was as follows:

verb noun prep unknown-noun

None of these data can be disambiguated by using the conventional best-matching algorithm, since noun2 does not appear in the example-base or thesaurus. Conjunctive relationships, described in Chapter 3, are used with the example-base and the thesaurus.

The results of the disambiguation are shown in Fig. 3. We were able to disambiguate 52.4% of the test data by using unknown-word-matching. By using Heuristic-1 in addition, we obtained a 72.4% success rate for unknown words.

One cause of failure is imbalance among examples. The number of examples for frequent verbs is larger than the number of examples for frequent nouns. As a result, verb attachment tends to be preferred.[2] Another cause of failure is the number of context dependencies. In the experiment, at most the nearest eight sentences were used; the optimum number is still an open question.

6 Conclusion

Methods for improving the coverage of example-bases were proposed in order to allow the realization of broad-coverage example-based systems. We are evaluating our approach with larger amounts of data. For future progress, the following issues must be discussed:

1. In this paper, conjunctive relationships were used as knowledge with the best-match algorithm, in addition to a thesaurus. However, various types of knowledge will be required on a large scale for a more robust system. Automatic or semi-automatic acquisition, using corpus-based methods, is also needed.
2. If there are many unknown words in an ambiguity, unknown-word matching will not work well. In addition to scaling up the example-base and the thesaurus, we should develop a more robust algorithm.

References

[1] Collins. *The New Collins Thesaurus.* Collins Publishers, Glasgow, 1984.

[2] IBM Corporation. *IBM Dictionary of Computing*, volume SC20-1699-07. IBM Corporation, 1988.

[3] S. Kinoshita, M. Shimazu, and H. Hirakawa. "Better Translation with Knowledge Extracted from Source Text". In *Proceedings of TMI-93*, pages 240–252, 1993.

[4] Merriam. *Webster's Seventh New Collegiate Dictionary.* G.& C. Merriam, Springfield,Massachusetts, 1963.

[5] K. Nagao. "Dependency Analyzer: A Knowledge-Based Approach to Structural Disambiguation". In *Proceedings of COLING-90*, pages 282–287, 1990.

[6] M. Nagao. "A Framework of a Mechanical Translation between Japanese and English by Analogy Principle". In A. Elithorn and R. Banerji, editors, *Artificial and Human Intelligence.* NATO, 1984.

[7] T. Nasukawa. "Discourse Constraint in Computer Manuals". In *Proceedings of TMI-93*, pages 183–194, 1993.

[8] S. Nirenburg, C. Domashnev, and D. I. Grannes. "Two Approaches to Matching in Example-Based Machine Translation". In *Proceedings of TMI-93*, pages 47–57, 1993.

[9] P. Procter. *Longman Dictionary of Contemporary English.* Longman Group Limited, Harlow and London, England, 1978.

[10] S. Sato and M. Nagao. "Towards Memory-Based Translation". In *Proceedings of COLING-90*, pages 146–152, 1990.

[11] N. Uramoto. "Lexical and Structural Disambiguation Using an Example-Base". In *The 2nd Japan-Australia Joint Symposium on Natural Language Processing*, pages 150–160, 1991.

[12] N. Uramoto. "Example-Based Word-Sense Disambiguation". *IEICE Transactions on Information and Systems*, E77-D(2), 1994.

[2] We did not use other heuristics such as preference for inner attachment.