Sentence disambiguation by document oriented preference sets

Hirohito INAGAKI, Sucharu MIYAHARA, Tohru NAKAGAWA, and Fumihiko OBASHI
NTT Human Interface Laboratories
1-2356, Take , Yokosuka-Shi,
Kanagawa,
238-03, JAPAN
E-mail: inagaki%ntthlt.NTT.JP@relay.cs.net

NTT Intelligent Technology Co.,Ltd.
223-1,Yamashita-Cho, Naka-Ku,
Yokohama
231, JAPAN

Abstract:

This paper proposes document oriented preference sets(DoPS) for the disambiguation of the dependency structure of sentences. The DoPS system extracts preference knowledge from a target document or other documents automatically. Sentence ambiguities can be resolved by using domain targeted preference knowledge without using complicated large knowledgebases. Implementation and empirical results are described for the analysis of dependency structures of Japanese patent claim sentences.

1. Introduction

Ambiguity in sentence interpretation is a major problem in natural language processing(NLP). Conventional NLP systems often use ad hoc or extremely large knowledgebases (pragmatic / semantic / commonsense) to eliminate ambiguities. Such systems are too slow and sometimes provide incomplete analyses. They have the further handicap that very large knowledgebases are needed. Asking the user for confirmation [Nishida 1982] is a practical solution to get correct parse-trees, but this confirmation is not useful for further computations. A practical NLP system should produce accurate results automatically while using a simple method and simple knowledge.

Preference models [Petitpierre 1987, Fass 1983, Schubert 1984], such as preference semantics, scoring, and syntactic preference are good candidates for a practical NLP system, because these models utilize simple ready-made knowledge like semantic markers or case frame dictionaries. The most difficult problem with preference models is the selection of the most appropriate preference knowledge that will induce a correct interpretation. However, preference knowledge extracted from a large corpus or an on-line dictionary [Jensen 1987] induces preference knowledge conflicts which block complete disambiguation.

To solve this problem, we introduce Document oriented Preference Sets(DoPS). The concept of DoPS is that to determine the most appropriate preference knowledge, preference knowledge be segregated into several domains, for example, language domain, field domain, and sentence domain, each of which has a different execution priority. By using the segregated preference knowledge in the fixed order, the most plausible interpretation can be obtained more rapidly and more accurately.

2. The concept of DoPS

Syntactic rules are capable of producing many sentence parse-trees. These parse-trees are syntactically correct, but most are incorrect from the view points of semantic meaning, contextual meaning, common-sense, specific field knowledge. It is necessary to use appropriate knowledge (semantic / contextual / commonsense / specific field) to eliminate the incorrect interpretations. For example, consider passage 1 of Figure 1. There are two possible interpretations for the gerund-phrase attachment.

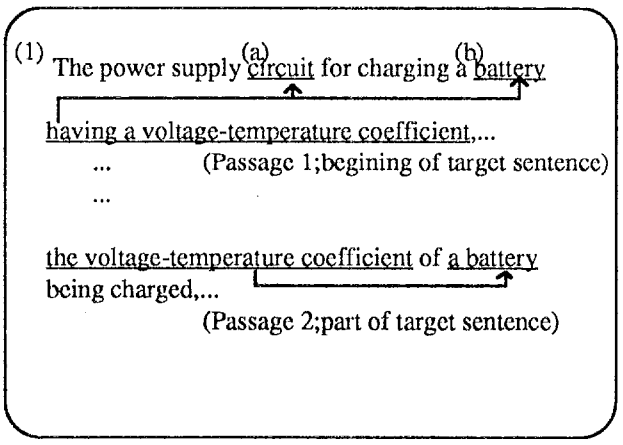

Figure 1. Example

People with electrical-engineering knowledge know that batteries have voltage-

temperature coefficients, not circuits. However if specific field knowledge is lacking, it is difficult to determine which is correct.

The notion of the DoPS is to utilize preference knowledge which can be extracted from other sentences of the target document or other documents. Documents sometimes contain paraphrases and the same or similar expressions. These expressions can contain several kinds of knowledge (semantic / contextual / commonsense / specific field). Sentence disambiguation can be based on such knowledge. For example, from passage 2 (which was written in another part of the target sentence(1)), it is clear that the coefficient of voltage-temperature is a property of the battery, thus the beginning of sentence (1) can be disambiguated.

This notion will be useful for any NLP stage, but it will be especially useful for dependency structure analysis. A DoPS is a collection of plausible combinations of phrases or words. To eliminate conflicts of preference knowledge, a hierarchical structure of preference knowledge is adopted in the DoPS. Figure 2 shows a hierarchical structure of a DoPS. The domains are, in order of increasing priority, language, application, field, author, document, paragraph, and sentence.

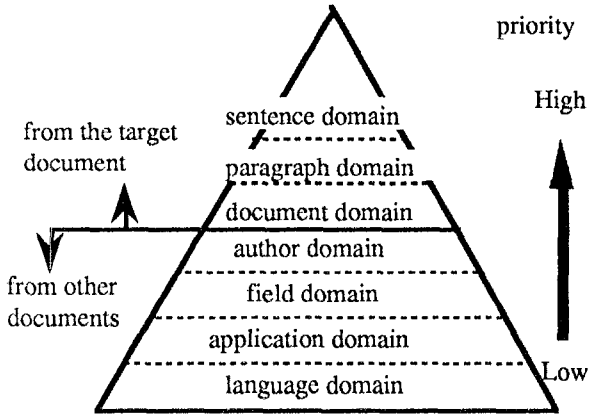

Figure 2. A hierarchical structure of DoPS

The language domain in a DoPS contains general language preference extracted from a large database, such as a word corpus or on-line dictionary. In the application domain (e.g. patent claim sentences, news papers, editorials, manuals), there exists application dependent phrases or word relations. In the field domain (e.g. electrical engineering, chemistry, agriculture), there exists field specific phrases or word relations. The author domain include author's characteristics as shown in his writing. A author often write on several documents in the same field. We consider that the knowledge associations held in the document, paragraph, sentence domains are more reliable than those in other domains.

DoPS entries of document, paragraph, sentence domains are acquired from the target document during the analysis, others can be prepared before analysis. For example, in Figure 3, if the author of document B is the same as document A, same DoPS entries of author, field, application, language domains are used in the analysis. Other domains, that is, sentence, paragraph, document domain are acquired during the analysis.

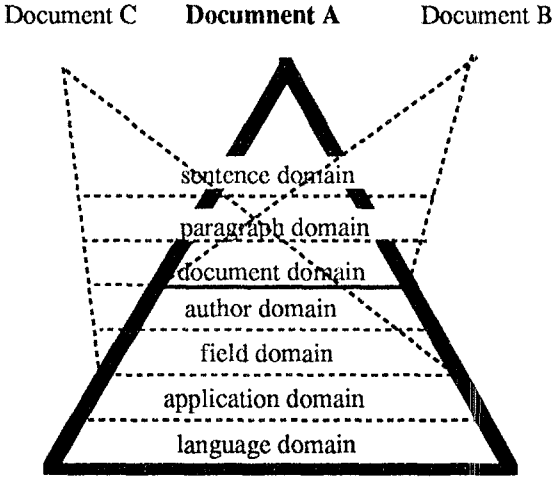

Figure 3. Different structures of DoPS

By using such domain structured preference knowledge, the system can extract the most plausible interpretation.

Figure 4 shows DoPS system flow diagram. First, the system starts analyzing the dependency structure of the target sentence with conventional syntactic rules. From each confirmed dependency relation, DoPS system develops a knowledge association or entry.

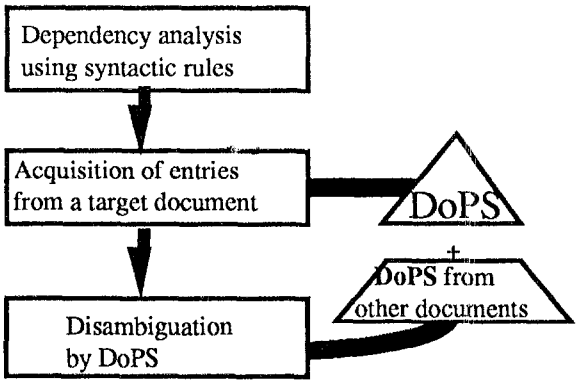

Figure 4. DoPS system flow diagram

In Figure 1 , since passage 2 is disambiguous, the DoPS system extracts two entries (Entry 3 and Entry 4) listed below.

Passage 3:
the voltage-temperature coefficient of a battery
Entry 3:
(voltage-temperature coefficient) [of,verb(passive)] (battery)
{sem:85 NUMBER} {sem:160 POWER}

Passage 4:
a battery being charged
Entry 4:
(battery) [nil,BE(passive)] (charge)
{sem:160 POWER} {sem:54 STORAGE}

sem: thesaurus category number
(e.g. Roget's thesaurus)
() : independent word , [] : intermediary

The DoPS entries are similar to the dependency relationships in dependency grammar, but two expansions have been made:

-semantic expansion
-coordination expansion

Semantic expansion ensures that for efficient use of DoPS, the dependency relationships will be expanded into semantic dependency relationships. In passage 3, the entry 3 is extracted as a dependency relation between instances. These will be semantically expanded by using an ordinary thesaurus dictionary(e.g. Roget's thesaurus). For example, the thesaurus category number of "battery" is 160 and the broader-word is "POWER". This means the word "battery" is a member of a word group named "POWER". This word group contains "power pack", "charger", "condenser", and so on. It is assumed that the same dependency relation will be valid for other members of the same word group. Passage 5 can be validate by entry 3 from passage 3.

passage 5:
This *condenser* is charged automatically.
{sem:160 POWER} {sem:54 STORAGE}
;"condenser" is the same word group as
;"battery"

The other expansion is to exchange the intermediary expressions (usually pre-positional words or verb). The transformation rules of intermediary expressions will be written in the DoPS system like [nil,BE(passive)] = [BE(passive)]<--> [nil].

Passage 6:
I charged this new battery yesterday.
Entry 6:
(charge) [nil] (battery)

Coordination expansion means that a DoPS like preference sets can be constructed using coordinated relationships between the coordinated sentence constituents. Using the coordinated constituents of preference sets, ambiguous constituents can be uniquely resolved, if the same type of coordinated sentence exists somewhere else in the target document or other documents.

In passage 7, it is clear that "records" and "files" is coordinated constituents. Preference sets for coordinated constituents is extracted as Entry 7. Using entry 7, the coordination in passage 8 is disambiguated.

Passage 7:
Were records and files dumped?
Entry 7:
(record) [and] (file)

Passage 8:
Old records and files were dumped.
Coordination:
*(Old records) [and] (files) were dumped.
Old (records) [and] (files) were dumped.

Even when semantically expanding the disambiguous dependency relations, ambiguities sometime persist. If ambiguous parts remain, the system adds ambiguous entries to the DoPS. In any domain, the execution priority of disambiguous entries is, of course, higher than that of ambiguous entries. Thus the target candidate is analyzed with disambiguous entries first. After that, if ambiguities still persists, the ambiguous entries are used.

Finally deterministic rules, such as right association or minimal attachment, must be used to eliminate any remaining ambiguity.

3. The DoPS system for Japanese dependency analysis

In this section, we describe the implementation of the DoPS system of Japanese dependency analysis.

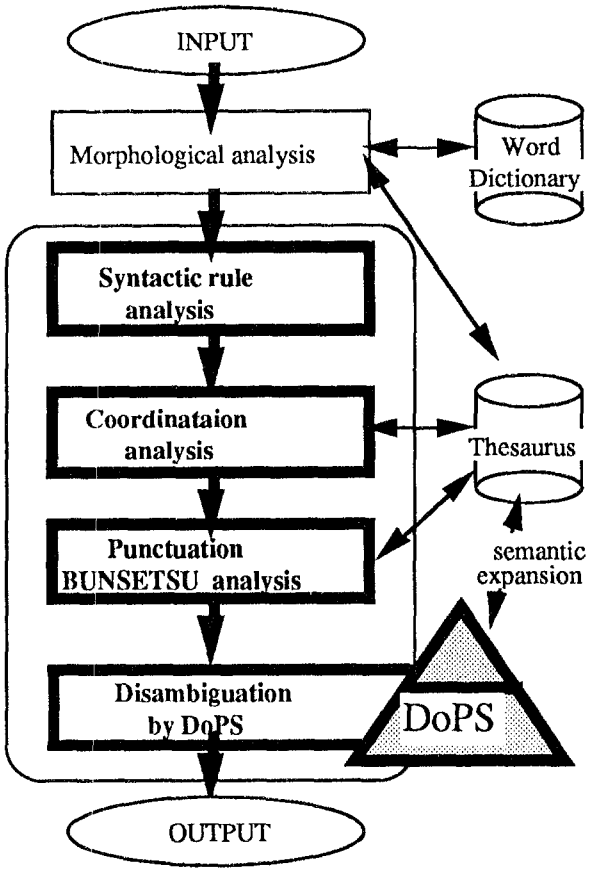

Figure 5. DoPS system for Japanese dependency analysis.

A DoPS system was implemented for Japanese dependency analysis and, because patent claim sentences have a tendency to use many similar expressions, the target documents were Japanese patent claim sentences. The implemented system restricted the application domain to patent claim sentences and activated only the application and higher domains. Figure 5 shows the implemented system. If dependency analysis using syntactic rules can resolved all sentence ambiguities, execution was stopped and DoPS entries were not created.

The syntactic rules used here were the general dependency rules and affiliated-word rules. The general dependency rules are (1) dependency relationships must not cross and (2)each verb doesn't have same case. The affiliated-word rules are given in table 1 which represents the connection between the governor and the dependant. In Japanese, the governor is the word units, BUNSETSU, which modifies another BUNSETSU, called the dependant. The properties of governor can be determined from the last post-positional word and are dependant on the last independent word in the BUNSETSU.

Table 1. Example of affiliated-word rules

dependant / governer	NOUN	VERB	ADJ(ADV)
post-positional "ga","ni","de"	NO	YES	YES
post-positional "wo","he"	NO	YES	NO
post-positional "no"	YES	NO	NO

"YES","NO" :connectivity of governer and dependant

The acquisition of DoPS entries begins after syntactic analysis is completed. The system analyzes the sentence structure within a document and chooses the disambiguated parts as entries as well as converting all dependency relationship candidates into ambiguous entries. For example, if the system executes syntactic analysis and finds passage 9 disambiguous, then the acquisition process creates entry 9.

Passage 9:
Japanese: Kana-kanji henkan wo okonau.
(English: Performs kana-to-kanji conversion.)

Entry 9:
(Japanese):
(kana-kanji) + [nil, no] +(henkan)
(kana-kanji henkan) + [wo] + (okonau)
(English):
(kana-to-kanji) + [/of] + (conversion)
(perform) + (kana-to-kanji conversion)

"/" indicates that this can be used in reversed relationships.

After all entries are extracted from the target document, the system executes coordination analysis. The constituents are picked up using the similarity of constituent and conjunction "to", "ya", and "mataha" as a clue. If the coordination analysis fails to eliminate all ambiguity, constituents are determined from coordinated constituents of preference sets.

After coordination analysis is completed, punctuation BUNSETSU analysis starts. In patent claim sentence, punctuation marks are used mainly for a restriction of the nearest dependency relation not for emphasis.

Finally, disambiguation of the dependency structure is commenced. In the disambiguation process, first the disambiguous entries are compared against the ambiguous parts of the sentence. The most similar

dependency relation is selected as the correct relation. During the disambiguation process, disambiguated knowledge associations are added to the DoPS. If there are many candidates of similar relations, the highest scoring candidate is selected. In one domain, first disambiguous then ambiguous entries are applied. The Japanese deterministic rule to is to choose the nearest dependency relation. Using this rule, all ambiguous relations will be disambiguated.

4. System empirical results

To test the effectiveness of the implemented DoPS system, we analyzed 10 real Japanese patent claim sentences; a total of nearly 7,000 words. These sentences were randomly selected from the computer and control systems region (the International patent classification G06F).

Only half of the dependency relations will determined before the disambiguation by DoPS. After the disambiguation by DoPS performed, we obtained an averaged accuracy of 93%(accuracy is defined as the number of right dependency relationships / the number of dependency relationships). Finally by using the deterministic rule, we obtained an averaged accuracy of 97 %. A simple system, using only deterministic rule, can obtain the average accuracy only 84%. Compared to this simple system, the sentence dependency analysis of our DoPS system can disambiguate with a high degree of accuracy, without needing a large knowledgebase.

In this experiment, most errors occurred during coordination analysis and disambiguation. Therefore, it is necessary to resolve coordination problems and to achieve more accurate disambiguation with DoPS. A more accurate DoPS system requires the elimination of useless and wrong entries. In the DoPS disambiguation process, utilization of dependency relations from case frame dictionaries is also needed.

5. Conclusion

We have described a new dependency structure analysis method using document oriented preference sets. The DoPS system extracts plausible preference knowledge from the target document or other documents.

Using a DoPS system for Japanese dependency analysis, we obtained an average accuracy of 97%. Compared to the 84% accuracy of simple analysis, it is clear that DoPS is more accurate. Furthermore, the concept of DoPS can also be applied to other NLPs such as MT [Tanaka 1990].

5. References

Fass,D. and Wilks,Y. "Preference Semantics, Ill-formedness, and Metaphor", Am. J. of computational Linguistics, Vol.9, No.3-4,pp.178-187,1983.

Jensen,K. and Binot,J. "Disambiguating prepositional phrase attachments by using on-line dictionary definitions" Computational Linguistics, Vol 13, No.3-4, pp.251-259, 1987.

Nishida,F. and Takamatsu,S. "Structured-information extraction from patent-claim sentences" Information Processing & Management, Vol.18, No.1, pp. 1-13,1982.

Petitpierre,D. ,Krauwer,S. , Arnold,D. ,and Varile,G.B. "A model of preference" ,Third conference of the European chapter of the Association for Computational Linguistics,pp.134-139, 1987.

Schubert, L. K. "On parsing preferences", Proceeding of COLING'84,pp.247-250,1984.

Tanaka,K., Nogami,H., Hirakawa,H., Amano,S., "Machine translation system using information retrieved from the whole document" 40th-Johoshorigakkai,pp.405-406,1990.