Recognizing Coordinate Structures for Machine Translation of English Patent Documents*

Yoon-Hyung Roh, Ki-Young Lee, Sung-Kwon Choi, Oh-Woog Kwon, and Young-Gil Kim

Natural Language Processing Research Team, Electronics and Telecommunications Research Institute,
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, Korea
{yhroh, leeky, choisk, ohwoog, kimyk}@etri.re.kr

Abstract. Patent machine translation is one of main target areas of current practical MT systems. Patent documents have their own peculiar description style. Especially, abstracts or claims in patent documents are characterized by their long and complex syntactic structures, which are often caused by coordination. So, syntactic analysis of patent documents requires special treatment for coordination. This paper describes a method to deal with long sentences in patent documents by recognizing coordinate structures. Coordinate structures are recognized using a similarity table which reflects parallelism between conjuncts. Our method is applied to a practical MT system and improves its quality and efficiency.

Keywords: Machine Translation, Patent Document, Coordinate Structure, Syntactic Analysis

1. Introduction

Patent machine translation is one of main target areas of current practical MT systems such as the English-Korean patent machine translation system (Kwon, 2007). Patent documents have their own description style and there have been some studies on the analysis of patent documents (Sheremetyeva, 2003; Shinmori and Okumura, 2003; Shinmori and Okumura, 2002). Especially, abstracts or claims in the patent documents are notorious for their long and complex syntactic structures, which are usually caused by coordination or relative clauses. Long sentences formed by relative clauses can be handled by segmentation (Kim et al, 2001). On the other hand, in case of long sentences formed by coordination, segmentation can cause syntactic analysis errors, because a segment resulting from segmentation can be dependent on the other constituents in the parse tree. Also, coordinate structures in patent documents have usually a large number of coordinate conjuncts (which we will call nodes) and can cause syntactic ambiguity explosion and parsing failure at the worst case in practical MT systems. So, syntactic analysis of patent documents requires special treatment for coordination.

There have been many computational researches about coordinate structures (Kaplan and Maxwell, 1988; Kosy, 1986). But, it is unrealistic to apply most of them to large-scale MT systems as mentioned in (Okumura and Muraki, 1994; Kurohashi and Nagao,. 1994). The more practical approaches about analyzing coordinate structures such as (Okumura and Muraki, 1994;

* This work was funded by the Ministry of Information and Communication of Korean government.

Kurohashi and Nagao, 1994; Agarwal and Boggess, 1992) all analyze coordinate structures using parallelism between conjuncts, but they are mainly targeted to recognizing two conjuncts (i.e., pre-conjuct and post-conjuct). Out of them, (Kurohashi and Nagao,. 1994) is one of the most practical approaches. They recognize coordinate structures in a Japanese sentence by constructing a similarity matrix between bunsetsus and searching a path with the highest parallelism in the similarity matrix using a dynamic programming method. However, that method is inadequate to apply to patent documents which have usually a large number of coordinate nodes and sometimes complex modification such as an inserted clause. We devised an appropriate method to recognize coordinate structures for patent documents using a similarity table. Although our method seems similar to that method in appearance, it is considerably different from that in the manner of constructing a similarity table and finding coordinate structures. Our method is simpler but more effective in patent documents.

In the next section, we outline the characteristics of coordinate structures in patent documents. And in the section 3, we present a method to recognize coordinate structures. In the section 4, we show experimental results and some analysis of the erroneous results, and then conclude our paper with some future works.

2. Coordination in Patent Documents

Figure1 shows a typical sentence in an abstract of a patent document. This sentence belongs to enumeration in the patent description style, and describes elements of a product. By analyzing many example sentences in patent abstracts, we can outline the characteristics of the enumeration sentences of the patent abstracts as follows:

1) They often have some keywords such as “include, comprising, having, step_of, unit, means”
2) In case of enumeration of noun phrase (NP), the definite article such as “the, each, said” is not used in the head node of NP. The definite article is usually used as elaboration.
3) They generally follows the normal form “ X (, X)* (,) and X”, where “()” means optional, and “*” means any number of repetition.

In this paper, on the basis of above features, we describe a method to recognize coordinate structures especially corresponding to enumeration in patent abstracts.

> A machine translation and telecommunications system includes a machine translation engine for translation of input text from a source language to a target language, a dictionary database including a core dictionary and a plurality of sublanguage (domain) dictionaries usable for translation from a source to a target language, a receiving interface for receiving text input from any of a plurality of users, each text input being accompanied by control information including user ID data indicative of one or more sublanguages preferred by a particular user, an output interface, and a dictionary control module coupled to the receiving interface responsive to the user ID data indicative of a sublanguage preference of a particular user for selecting a corresponding sublanguage dictionary of the dictionary database to be used by the machine translation engine along with the core dictionary for performing translation of the particular user's text input.

Figure 1: An example sentence of patent abstract

3. Recognizing Coordinate Structures by Similarity Table

As mentioned above, an enumeration sentence usually enumerates many elements or procedures and each element or procedure can have modifiers and nested coordination. So, overall sentence structure can be excessively complicated and the process of recognizing coordinate structures can be difficult. For this, we simplify the analysis target by recognizing all possible coordinate nodes and construct a similarity table for using parallelism between the coordinate nodes.

3.1.Recognizing Possible Starting Points of Parallel Nodes

Recognizing coordinate structures is carried out after mophological analysis, tagging and base NP(BNP) chunking. The figure 2 represents the result of BNP chunking of the example

sentence. Assuming that all of coordinate structures follow the normal form "Xs (, Xm)* (,) and Xe", there are three types of nodes, which are a starting node(Xs), a middle node(Xm), and an ending node(Xe). The starting points of coordinate nodes are recognized by some syntactic cues. By corpus analysis, we find that the syntactic tags of coordinate structures are mainly a noun phrase(NP), a verbal phrase(VP), and a that-clause(SBAR). The syntactic cues for recognizing starting points of coordinate nodes are shown in the table 1.

Table 1: The syntactic cues for recognizing the starting points of coordinate nodes.

Node Tag	Starting Node	Middle Node	Ending Node
NP	PREP|VERB /BNP	, /BNP	(,) and /BNP
VP	PREP|VERB /(ADV) VBG	, /VBG	(,) and /VBG
SBAR	PREP|VERB /that	, /that	(,) and /that

In the table 1, "VBG", "VERB", "PREP", and "ADV" repesents a verb with ing-form, a verb, a preposition, and an adverb respectivey. And '|' means "or" and '/' means the starting point of a coordinate node. In the figure 2, the mark '/' represents the candidate starting points of the coordinate nodes recongnized by the given syntactic cue.[1]

In case that there comes a main verb before a comma or the end of the sentence after the ending node of a NP coordinate structure, we exclude that point from the starting points of the candidate coordinate nodes, regarding it as a starting point of a new clause.

[A machine translation] and [telecommunications system] includes /[a machine translation engine] for [translation] of [input text] from /[a source language] to /[a target language], /[a dictionary database] /including /[a core dictionary] and /[a plurality of sublanguage (domain) dictionaries] usable for [translation] from /[a source] to /[a target language], /[a receiving interface] for /receiving [text input] from /[any of a plurality of users], [each text input] /being accompanied by [control information] /including [user ID data] indicative of /[one or more sublanguages] preferred by /[a particular user], /[an output interface], and /[a dictionary control module] coupled to [the receiving interface] responsive to [the user ID data] indicative of /[a sublanguage preference] of /[a particular user] for /selecting /[a corresponding sublanguage dictionary] of [the dictionary database] to be used by [the machine translation engine] along with [the core dictionary] for /performing [translation] of [the particular user]'s [text input]

Figure 2: The result of BNP chunking and recognition of starting points of coordinate nodes

3.2.Constructing Similarity Table

We construct a similarity table between the recognized nodes in order to use parallelism between the coordinate nodes. In the similarity table, the value of i-th row and j-th column(Si,j) means the similarity between the i-th candidate node and the j-th candidate node. The similarity between nodes is composed of the head node simiraity(s0), the head word similarity(s1, s3), and the structural similarity(s2). Then, Si,j is calculated as follows:

$$Si,j = s0*(s1 + s2 + s3) \quad (1)$$

s0: tag similarity of the node (e.g., 1 if their tags are the same and are not NP, otherwise 1 if both of their tags are NP and their determiner types are compatible, otherwise 0, the determiner type will be explained later)

[1] For simplicity, we omit some starting points of NP nodes, which have no effect on the result.

s1: lexical or tag similarity of the head word (e.g., 6 if their lexicals are the same, otherwise 4 if their tags are the same and their node is not NP and 2 if their tags are the same and their node is NP, otherwise 0).
s2: lexical or tag similarity of the next word of the head word (e.g., 4 if their lexicals are the same, otherwise 2 if their tags are the same, otherwise 0)
s3: determiner similarity in case of NP (e.g., 5 if the determiner type is the same, otherwise 0)

There are four determiner types according to the determiner and the plurality of the head word. The determiner type 1 is the case where there is a indefinite article like “a machine translation engine”, the determiner type 2 is the case where there is no determiner and the head word is plural like “one or more sublanguages”, the determiner type 3 is the case where there is no determiner and the head word is singular like “translation”, and the determiner type 4 is the case where there is a definite determiner like “the receiving interface”. The quantifier such as “a plurality of” is considered as the case having no determiner. The determiner type 1 and 2 are compatible with each other. The determiner type 2, 3, and 4 are compatible with one another. We exclude the nodes with the determiner type 3 and 4 as the final coordinate nodes because the element of a product is a common noun and they don’t have a definite article as described in the second characteristics of enumeration sentences in the section 2.

The simiarity values assigned above just reflect the rough priority between features with the order of the node tag similarity, the head word similarity, and the structural similarity. Also, the lexical similariy has precedence over the tag similarity. The specific values need to be decided by exeriment.

The figure 3 shows the similarity table of the example sentence. The node number is the sequential number of the recognized coordinate nodes, and the chunk number is the sequential numer of the chunks resulted from the base NP chunkng. The symbols ‘ <’, ‘ ,’ , ‘ >’ in the i-th row and the i-th column represent a starting node, a middle node, and an ending node repectively.

For example, S6,4 is the similarity between the node starting with “a receiving interface for” and the node starting with “a core dictionary and”, and the similarity value is as follows:

s0 = 1, s1 = 2, s2 = 0, s3 = 5 → S6,4 = 2 + 0 + 5 = 7

S6,1 is the similarity between the node starting with “a receiving interface for” and the node starting with “a machine translation engine for”, and the similarity value is as follows:

s1 = 2, s2 = 4, s3 = 5 → S6,1 = 2 + 4 + 5 = 11

3.3.Recognizing all Possible Coordinate Structures

All possible coordinate structures are recognized based on the similarity table. A coordinate structure is composed of one starting node, zero or more number of middle nodes, and one ending node. So, for a given ending node, we can generate a coordinate structure by first selecting a starting node having more than zero value of similarity with the ending node, and adding middle nodes having more than zero value of similarity with the ending node between the the starting node and the ending node.

Once a coordinate structure is identified, the scopes of all coordinate nodes are determined except the ending node. The scope of the ending node is not obvious in an English sentence, so temporarily we decide as its scope the minimum scope which can make any coordinate node. For example, in case of NP, its ending position of ending node is the ending position of first BNP and in case of VP, it is first VBG. Once all the scopes of the coodinate nodes are determined, then we make some simple checks whether the scopes of each nodes can be parsed to the correspoding node. The representative checking method is to check how many main verbs exist in a node. In case of NP, VP, and PP, there must be no main verb. Conversely, in case of SBAR, there must be a main verb. A main verb is a verb which have tense and so can form a clause. In case that there is relative clause, we subtract the number of relatives from the number of main verbs. If there is any node which doesn't satisfy that constraint in a coordinate structure, that coordinate structure is excluded.

node No.	chunk No.	head word	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	8	engine	<	7	0	7	7	11	0	0	0	7	11	7	0	9	0
2	26	database		,	0	7	7	7	0	0	0	7	7	7	0	7	0
3	29	including			<	0	0	0	4	4	11	0	0	0	8	0	4
4	30	dictionary				<	11	7	0	0	0	7	7	7	0	11	0
5	34	dictionaries					>	7	0	0	0	7	7	7	0	11	0
6	53	interface						,	0	0	0	7	11	7	0	9	0
7	57	receiving							<	4	6	0	0	0	4	0	6
8	71	being								<	4	0	0	0	4	0	4
9	76	including									<	0	0	0	4	0	6
10	77	user										<	7	7	0	2	0
11	77	interface											,	7	0	2	0
12	97	module												>	0	7	0
13	122	selecting													<	0	4
14	123	dictionary														<	0
15	145	performing															<

Figure 3: Similarity table of the example sentence

3.4.Selecting the Final Coordinate Structure

Since it is assumed that all the coordinate structures follow the normal form "X (, X)* (,) and X", there are as many coordinate structures as the number of the ending nodes in a sentence. The scopes of those coordinate structures should not be crossed. In other words, one coordinae structure either have exclusive scope with the other coordinate structures or is entirely included in the other coordinate structures. We will call such a coordinate structure set a consistent coordinate structure set(CCS). All CCSs can be obtained by checking the no-crossing condition with all possible coordinate structure combinations. In nested coordinate structures, we constrain the inner-most coordinate structure to have the narrowest scope, e.g. have the nearest starting node out of possible starting nodes as its starting node. The inner coordinate structure is considered as one node represented by the starting node when we count the coordinate nodes of the outer coordinate structure. So, we exclude the middle nodes in the outer coordinate structure overlapping with the middle nodes in the inner coordinate structure. As a result, there are 4 CCSs in the example sentence as shown in the figure 4.

The score of a CCS is calculated by the sum of the score of each coordinate structure in the CCS. The score of a coordinate structure is basically calculated by the sum of similarity values between the ending node and the other nodes. In addition to that, some weights according to context are added. Such weights can be given as follows:

1) 7, when the words prior to the starting node is "include | comprise | comprised_of"
2) 7, when the words prior to the starting node is "including | comprising | step_of | means_for"
3) 3, when the word prior to the starting node is "having"
4) 1, when the word prior to the starting node is a verb
5) 3, when the head word of NP is " unit | means"

The scores of the CCSs in the example setence is as follows:
CCS 1: (S5,1 + S5,2 + 7) + (S12,11+S12,10) = 7 + 7 + 7 + 7 + 7 = 35
CCS 2: (S5,4 + 5) + (S12,10 + S12,11) = 7 + 5 + 7 + 7 = 26
CCS 3: (S5,4 + 5) + (S12,1 + S12,2 + S12,6 + S12,11 + 7) = 7 + 5 + 7 + 7 + 7 + 7 + 7 = 47
CCS 4: (S5,4 + 5) + (S12,4 + S12,6 + S12,11) = 7 + 5 + 7 + 7 + 7 = 33

Then we select the outer-most coordinate structures in the CCS with the highest score as the final coordinate structure. In case of the example sentence, the CCS 3 is selected as the CCS having the highest score, and the coordinate structure (1,2,6,11,12) is selected as the final result. In the real cases, we eliminate the final coordinate structure with very small scope corresponding to a very local coordinate structure.

Then, all the nodes of the recognized coordinate structure are parsed, and the whole coordinate structure is reduced to one node. The final parsing result is produced by parsing the whole sentence, with the recognized coordinate structure substitued by that node.

However, there is a problem that the ending point of the ending node is ambiguous, because it is not guranteed that the ending point of the ending node be always the end of the sentence. For this, the scope excluding the ending node is recognized as the scope of the coodinate structure, and reduced to one node, thus leaving the decision of the adequate scope made by parsing.

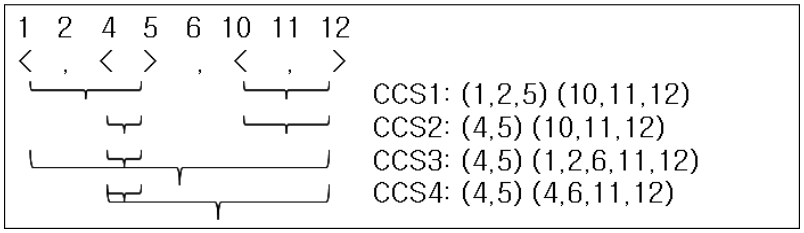

Figure 4: All Possible CCSs in the example sentence

4. Experimental Results

For experiment, we extracted 840 patent abstracts from computer/electronics fields and applied our method to them. Out of 840 abstracts, the number of sentences which has actually coordinate structures is 94 sentences. The average sentence length is 107.1 words/sentence. The table 2 shows the precision and the coverage of our method. It shows relatively high precision but low coverage. Although the ratio of coordinate structures in the entire sentences of abstracts is not so high, considering the importance of an abstract in a patent document and the excessively long sentence length more than 100 words, the correct analysis of coordinate structures has important effect on the overall machine translation quality.

Table 2: Experimental result for recognizing coordinate structures.

Correct	Incorrect	precision	recall	F-measure
70	8	89.7%	74.5%	81.4%

The incorrect results are categorized as follows:

1) The input sentence itself is an erroneous sentence.
2) There is a coordinate structure, but it is not recognized because of low coverage of our method.
3) Incorrect recognition of starting nodes or middle nodes.
4) There is not a coordinate structure, but a coordinate structure is recognized.

The main causes of the errors are shown in the table 3. In many cases, the errors are caused by our assumption about coordination structures such as the form of a coordinate structure or the constraint by the determiner type. From the result, we need to extend the coverage by considering more various cases.

Table 3: Main causes of the errors.

Cause of Errors	Example	Frequency
Comma, PP, etc. are inserted	"determines, for each translation example, a similarity", "comprises, a semiconductor chip"	4
Errors by determiners	"the laminating a sacrificial layer", ", syntax analyzer for"	5
Ambiguous Starting Nodes	"A dictionary retrieval device is constructed by a conversion character definition form for providing group IDs for character subsets, a character-group …"	3
Uncovered Case	PP, VP(declarative), INFP, "NP VP , …, and VP" form, SBAR(which-clause)	6

5. Conclusion and Future Works

We presented a method to recognize coordinate structures occurring typically in patent abstracts. The coordinate structures are recognized by searching all CCSs and scoring all CCSs using the similarity table. The experiment shows our method is effective for recognizing coordinate structures in patent abstracts.

For the future works, first, the target node for recognizing coordinate structures have to be extend to other case such as PP, to-infinitive, or all other relative clauses. Second, we have to consider the case that is out of our assumption such as the form of coordinate structures or the constraint by the determiner type. Lastly, we think that more robust treatment about inserted phrases or clauses is needed.

References

Agarwal, R., and L. Boggess. 1992. A simple but useful approach to conjunct identification. *In Proceedings, 30th Annual Meeting of Association for Computational Linguistics*, pp. 15-21.

Kaplan, R. M., and J. T. Maxwell. 1988. Constituent coordination in Lexical-Functional Grammar. *In Proceedings, Eighth International Conference on Computational Linguistics*, pp. 303-305.

Kim, S.-D., B.-T. Zhang, and Y. T. Kim. 2001. Learning-based Intrasentence Segmentation for Efficient Translation of Long Sentences. *Machine Translation*, 16(3):151-174.

Kosy, D. 1986. Parsing conjunctions deterministically. *In Proceedings of the 24th ACL Conference*.

Kurohashi, S., and M. Nagao. 1994. A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, 20(4), pp. 507-534.

Kwon, O.-W., S.-K. Choi, K.-Y. Lee, Y.-H. Roh, Y.-G. Kim. 2007. English-Korean Patent Translation System: FromTo-EK/PAT. *MT Summit XI Workshop on Patent Translation*.

Okumura, A., and K. Muraki. 1994. Symmetric pattern matching analysis for English coordinate structures. *Proceedings of the fourth conference on Applied natural language processing*, pp. 41-46.

Sheremetyeva, S. 2003. Natural Language Analysis of Patent Claims. *Proceedings of ACL 2003 Workshop on Patent Corpus Processing Workshop.*

Shinmori, A., M. Okumura, Y. Marukawa, and M. Iwayama. 2003. Patent Claim Processing for Readability. *Proceedings of ACL 2003 Workshop on Patent Corpus Processing Workshop*.

Shnimori, A., M. Okumura,Y. Marukawa, and M. IwaYama. 2002. Rhetorical Structure Analysis of Japanese Patent Claims Using Cue Phrases. *Proceedings of the Third NTRCIR Workshop*.