Language, Information and Computation (PACLIC12), 18-20 Feb, 1998, 303-309

Syntactic Verifier as a Filter to Compound Unit Recognizer

Hanmin Jung, Sanghwa Yuh, Taewan Kim, Dong-In Park*
Systems Engineering Research Institute

This paper describes the combination compound unit (CU) recognizer with syntactic verifier using partial parsing mechanism. The recognizer finds all the CUs, combined concept including collocations, idioms, and compound nouns, in input sentence. CU information reduces the search space of syntactic analysis and a portion of Part-Of-Speech (POS) ambiguities. Syntactic verification is to obtain precise CU recognition results by means of pruning wrongly recognized units that are caused by improper variable hypotheses. The experimental results show the precision of CU recognition is increased to 99.69% with 31 CFG rules on cyclic trie structure for 1,268 WSJ articles in the Penn Treebank. They also show CU recognition increases the understandability of translation for Web documents.

Keywords: compound unit (CU), syntactic verification, partial parsing, cyclic trie

1. INTRODUCTION

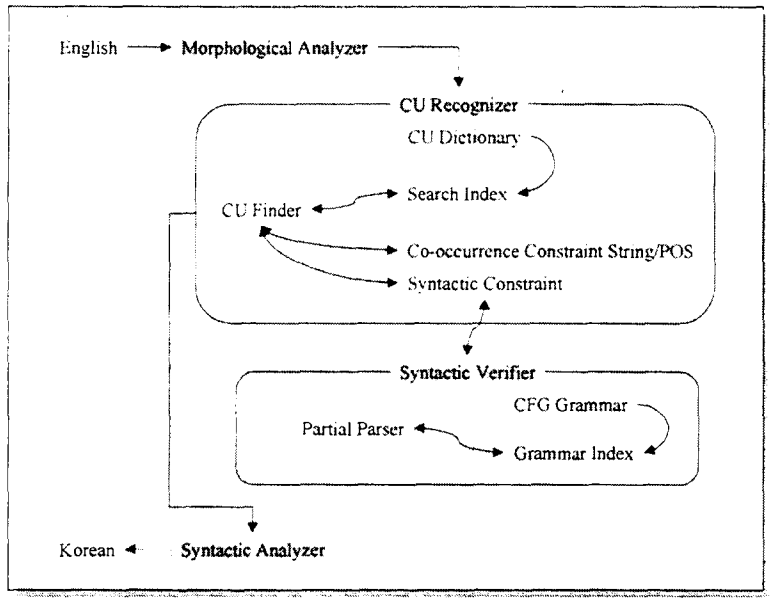
One of the problems of rule-based translation using only syntactic grammar rules is the difficulty to process the fixed expressions which frequently occur in the context [Bond *et. al.*, 1995] [Katoh & Aizawa, 1995] [Lauer & Dras, 1994] [Schenk, 1986]. "Mary keeps up with her brilliant classmates." and "I prevent *him* from going there." are the simple examples for the problem. Word-to-word translation would not process or naturally translate the expressions which cause a lot of complex and ambiguous parse tree structures [Yoon, 1994] [Yosiyuki *et. al.*, 1994].

There are many studies to resolve the above problem [Bond *et. al.*, 1995] [Lauer & Dras, 1994] [Li, 1995] [Yosiyuki *et. al.*, 1994]. A solution of for it considers a fixed pattern on the context as a translation unit. This method reduces the load of syntactic/morphological generation with pre-translated natural equivalents, the number of parse trees, and syntactic ambiguities by scaling down the search apace of syntactic analysis [Lee, 1994a]. A translation unit consists of a simple form with only fixed words and a complex one that includes variable word/phrase/clause in it. However, the studies have interest only on a part of the categories of the units, e.g. compound nouns or phrasal verbs. [Schenk, 1986] tries to get a solution for pattern recognition within a theoretical framework, but he does not consider the syntactic relation between the pattern and its neighbor. However, the relation is the crucial key for machine translation system to naturally translate a phrase/clause/sentence with pattern-based approach, especially in the case of the translation between SVO linguistic structure (e.g. English, German) and SOV one (e.g. Korean, Japanese). One of the global solutions for pattern-based approach is to find and apply all possible bilingual or multilingual fixed expressions as translation units. We define the compound unit (CU) as a combined concept including collocations, idiomatic expressions, and compound nouns [Jung *et. al.*, 1997a] [Jung *et. al.*, 1997b]. The combination of pattern-based approach using CU and rule-based translation makes our translator more tractable and adaptable for the open domains which have various sentence types in World Wide Web (WWW). This pattern-based module finds all compound units and their information between morphological and syntactic analyzer.

However, CU recognizer has a problem that it can not use any syntactic information to check syntactic constraints in CUs, but only surface form or its conjugation matching. It causes the degradation of the recognition reliability (precision). We propose the combination CU recognizer with syntactic verifier using partial parsing mechanism. The verification is to increase the reliability of the recognizer by means of pruning wrongly recognized CUs. Partial parser operates on a cyclic trie and uses simple CFG rules for the fast recognition. The experimental results show that it increases the precision of CU recognition up to 99.69%.

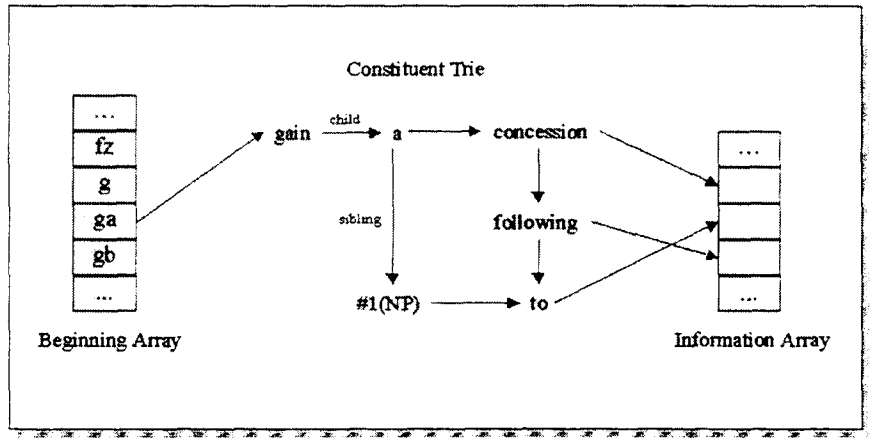
* Machine Translation Laboratory, Systems Engineering Research Institute, Eueon 1, Youseong, Taejeon, Korea, 305-333. E-mail: jhm@seri.re.kr

2. COMPOUND UNIT RECOGNIZER AS A PATTERN-BASED APPROACH



[Figure 2.1] The system structure of CU recognizer and its syntactic verifier

CU recognizer is a plug-in module located between morphological and syntactic analyzer. CU recognition reduces the search space of syntactic analysis and a portion of POS ambiguities. [Figure 2.1] shows the system structure of CU recognizer and its syntactic verifier. The recognizer finds all the types of CUs in input sentence on its search index using co-occurrence constraint string/POS and syntactic constraint. The index is automatically made from text-based CU dictionary.



[Figure 2.2] The search index of CU recognizer

Our search index structure consists of three parts: beginning array, constituent trie, and information array [Figure 2.2]. The beginning array is used for more rapid access to the constituent trie [Cho, 1992] [Fredkin, *et. al.*, 1960] [Knuth, 1973] [Lee, 1994b]. Each element of the beginning array corresponds to the first two characters in the first constituent of a CU. For example, "ga" is the element for all of the CUs that begin "gain." Empirically, in the case of using the first two characters instead of one, the number of the traverse on the index "methods" is reduced to 20 ~ 80%. The constituent trie is a modified trie structure with heterogeneous nodes: fixed (e.g. "gain", "a", "concession") and variable constituent (e.g. "#1 (NP)") nodes [Jung *et. al.*, 1997c]. A part of variable constituents have one or more conditions that consist of syntactic tags (e.g. NP, VP, NP-clause) and co-occurrence constraints (e.g. oneself = {myself, himself, themselves, ...}, one's = {my, his, their, ...}). The element of the information array has the whole information for a CU (e.g. representative POS tag, equivalent(s), and CU key). The array is to get the modularization of the index structure.

The principle of CU search is "most-specific-expression-first" [Yoon, 1994]. The traverse order on the siblings of the trie always keeps this principle. We provide three priorities for the traverse to apply our heterogeneous trie structure [Table 2.1]. The other search strategies on the structure are same as those of other ordinary trie structures.

[Table 2.1] The traverse order on the siblings of our heterogeneous trie structure

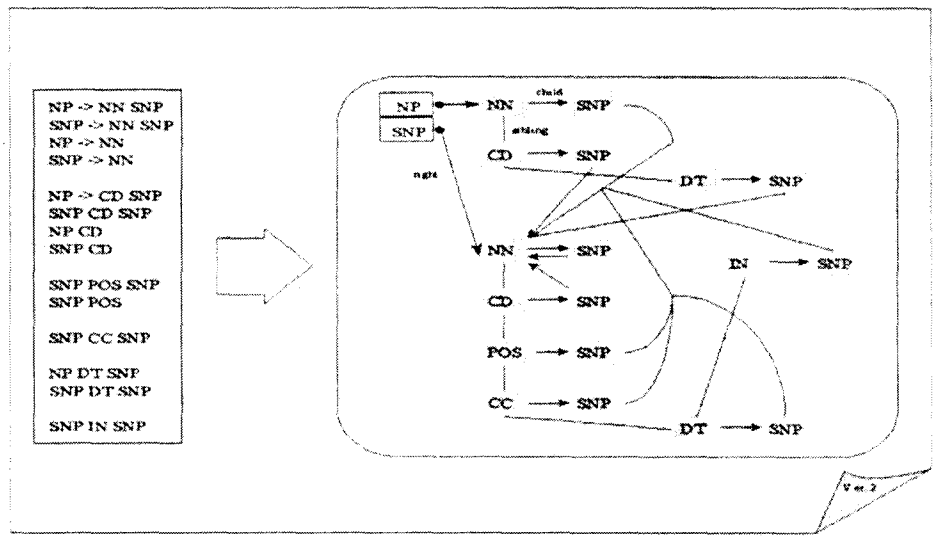
Priority	Node type	Internal traverse order
1	Fixed constituents	Alphabetic ascending order, Longer length first
2	Variable constituents with conditions	More conditions first
3	Variable constituent without any condition	Only one

3. SYNTACTIC VERIFIER USING PARTIAL PARSING

The syntactic verifier for the variable constituent that has one or more syntactic constraints overcomes the degradation of the precision for CU recognition by the absence of any partial syntactic analysis. For example, The recognizer would find a pseudo CU "take #1 (NP) to" in "But, it doesn't take *much* to get burned." by the absence. In the case of "Some researchers have charged *that administration is imposing new ideological tests* for top scientific posts", it can accidentally find a correct CU "charge #1 (NP-clause) for", but it is an unreliable result without any syntactic verification. We use a partial parser as the implementation of the verifier to raise the reliability of CU recognition. Its operation sequence is as follows.

(1) Load CFG rules.
(2) Partially parse in the manner of top-down using each of syntactic constraints (tags) in a selected CU hypothesis and a given sequence of POS tags from input sentence.
(3) Determine if the two are syntactically matched. In the case of matching failure, continue to match with the other syntactic constraints.
(4) Return the matching result to CU recognizer in order to accept the hypothesis.

The verifier is also able to partially parse the embedded syntactic structures, e.g. "... that Cray Computer anticipates needing perhaps another $120 million in financing beginning next September", with right recursion from the RHS of each grammar rule. We have a grammar index structure with cyclic trie for the recursion [Figure 3.1]. There are currently 3 LHS nodes and 38 RHS nodes for 31 CFG rules. We do not have any constraint except the sequence of the rule description because the partial parser is mainly focused on syntactically verifying with a fast and simple way.

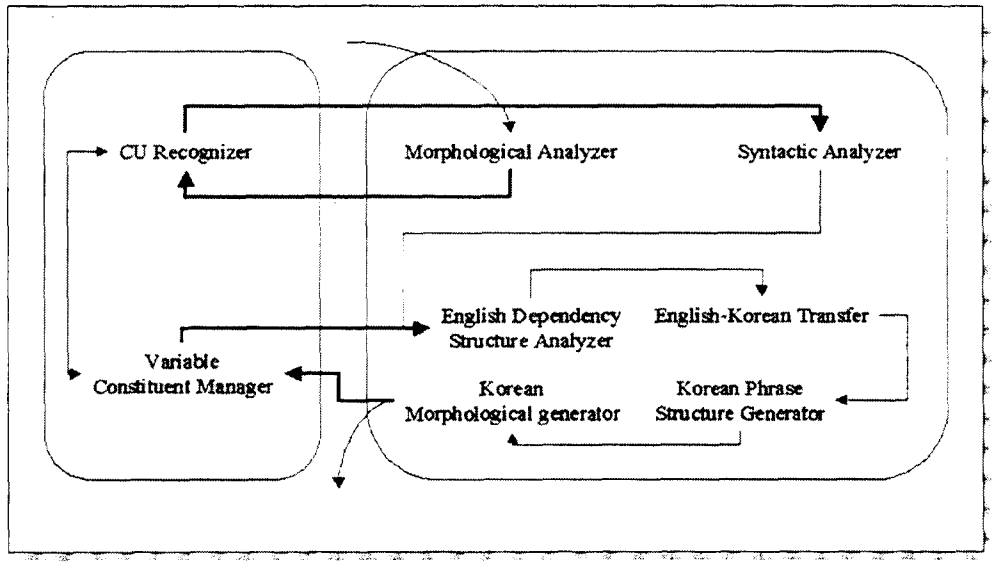


[Figure 3.1] The grammar index structure for syntactic verification

The verifier uses a look-ahead mechanism to match a node on the index with the current input POS tag while avoiding backtracking. The following pseudo codes are the main frame of the mechanism.

```
Look_ahead (RHS, tag) {
    If (RHS is non-terminal) {
        If (look_ahead (child, tag) is equal to TRUE)
            Return (TRUE);
        Else if (there is its sibling)
            Return (look_ahead (sibling, tag));
        Else
            Return (FALSE);
    }
    Else {
        If (RHS is equal to tag)
            Return (TRUE);
        Else if (there is its sibling)
            Return (look_ahead (sibling, tag));
        Else
            Return (FALSE);
    }
}
```

4. INTERFACE BETWEEN CU RECOGNITION AND RULE-BASED TRANSLATION



[Figure 4.1] The interface between pattern-based approach and rule-based translation
(Left box is for the pattern-based approach, and the right is for the rule-based translation)

CU recognition makes a problem of interfacing between pattern-based approach and rule-based translation. Ordinarily, rule-based parsers have a direct input from morphological analyzer without any combination with pattern-based modules. However, CU recognition as a pattern-based approach changes the whole or a part of POS sequence from input sentence by merging and replacing the POSs. To overcome this interface mismatch, We give additional features such as a representative POS, verb/adjective/noun types, a CU key, and a generation code to CU information. There are two cases that the representative POS of a CU is noun. First, simple compound noun such as "portfolio manager" does not need any additional features because the POS sequence of input sentence is not changed. Second, if the CU is a derived compound noun such as "ceiling (NP) on" or phrasal verb such as "talk about", it should have a feature to represent a relation between the verb/noun and its object(s). The feature is defined as the sub-categorization information of CU. There are two additional features; a generation code for the interface with generation module and a CU key which represents the representative

meaning of its CU to extract the most similar features from existing equivalent dictionary during English-Korean transfer without any additional word information.

In the case of variable constituents, although its syntactic verification is operated during CU recognition, whole parsing and transfer/generation is necessary to produce its equivalent. For the case, there are two operations; first, top-down parsing of syntactic analysis to get an equivalent of the constituent and second, insertion into the equivalent of its parent CU.

5. EXPERIMENTAL RESULTS

Our training corpus is WSJ 1,268 sentences in Penn Treebank [Marcus *et. al.*, 1993]. 1,194 CUs are manually extracted by linguists and about 6,800 compound nouns automatically extracted using specific POS sequence. The average word number of a sentence in the corpus is 15.33. Representative POSs for CUs are verb, noun, preposition, adverb, adjective, and sentence (e.g. It is ~ that ~). 56.26% of 1,636 recognized CUs are compound nouns, 29.58% are collocations and phrasal verbs. Recall is 97.65% and precision is 98.52% for the same corpus.

The combination CU recognizer with syntactic verifier increases the precision up to 99.69% from 98.52%; i.e. 77.78% of pseudo CUs are removed. 0.31% that are not pruned have the wrongly POS tagged words, e.g. "be expected to #1 (verb/noun)", "sell (verb/noun) #1 from." The followings are the examples of CU recognition results filtered and pruned by this verification.

(1) "Open #1 (NP) to" in "Sales in stores open *more than one year rose 3%* to $29.3 million."
(2) "Increase #1 (NP) by" in "Analysts attributed the increases *partly to the $4 billion disaster-assistance package enacted* by Congress."
(3) "Point #1 (NP) at" in "The FT 30-share index settled 16.7 points *higher* at 1738.1."

The time complexity of the syntactic verification using our cyclic trie is also linear bound as CU recognition by the following experiment. Due to the small size of CFG rules and trie nodes, the verifier little does not impose burden on CU recognizer. [Table 5.1] shows the linear relation between the number of CU and its recognition time. CU recognition with syntactic verification has little effect on the processing time when compared with the case of the recognition without syntactic verification. The way of CU dictionary loading preferably affects the time rather than other options.

[Table 5.1] The processing time of CU recognition with options
(SV: syntactic verification, DL: dictionary loading, IL: index loading)

CU	Option	Real time	User time	System time
2138	With SV and DL	17.1	7.9	7.3
	With SV and IL	11.3	2.6	7.0
	Without SV, with IL	11.2	2.6	6.9
3623	With SV and DL	21.5	11.9	7.5
	With SV and IL	11.4	2.7	7.2
	Without SV, with IL	11.3	2.7	7.2
5026	With SV and DL	25.5	15.4	7.9
	With SV and IL	11.6	3.1	7.0
	Without SV, with IL	11.6	3.2	7.0
6470	With SV and DL	29.9	20.0	7.6
	With SV and IL	12.0	3.5	6.8
	Without SV, with IL	12.0	3.5	6.8
7886	With SV and DL	34.7	24.2	8.3
	With SV and IL	12.2	3.5	7.0
	Without SV, with IL	12.1	3.4	7.0

We also experiment for our hybrid approach using CU recognition. We use the understandability [Choi & Kim, 1996] as a measure of the quality of translation. [Table 5.2] shows the degree of the understandability and [Table 5.3] the translation results of some Web homepages. We regard the degree 2, 3, and 4 as the criterion of acceptable translation. The translation results show that our hybrid mechanism increases the ratio of acceptable

translations to 86.8%. It implies that the pre-translated natural equivalents of CUs are helpful to understand target sentences.

[Table 5.2] The degree of the understandability

Degree	Meaning
4 (Perfect)	The meaning of the sentence is perfectly clear.
3 (Good)	The meaning of the sentence is almost clear.
2 (OK)	The meaning of the sentence can be understood after several readings.
1 (Poor)	The meaning of the sentence can be guessed only after a lot of readings.
0 (Fail)	The meaning of the sentence cannot be guessed at all.

[Table 5.3] The translation results of five Microsoft homepages
(The right percentage of the arrow on "Poor" degree means the rate of acceptable translation)

Item		Result	
The number of Sentences		273	
The number of words		1,693	
Average Number of word per sentence		6.2	
Degree		Before hybrid	After hybrid
The understandability	4 (Perfect)	37 (13.5%)	37 (13.6%)
	3 (Good)	45 (16.5%)	47 (17.2%)
	2 (OK)	148 (54.2%) -> 84.2%	153 (56.0%) -> 86.8%
	1 (Poor)	24 (8.8%)	17 (6.2%)
	0 (Fail)	19 (7.0%)	19 (7.0%)
Total number of sentences		273 (100%)	273 (100%)

5. CONCLUSION

Our rule-based translation method is a typical transfer-based one. The combination pattern-based approach using CU recognition with rule-based translation makes our translator more tractable and adaptable for the open domains that have various sentence types in WWW. The pattern-based module finds all pre-translated patterns and their information between morphological and syntactic analyzer.

The combination CU recognizer with syntactic verifier is to increase the reliability of the recognition. Experimental results show the precision increases to 99.69%. Partial parser as an implementation of the verifier syntactically checks the variable constituents in CUs by the use of simple and right-recursive CFG on our cyclic trie structure. There is a trade-off between processing time and exquisiteness for the recognition, however we are focusing to build a plug-in module with fast and simple syntactic verification. The results also show our hybrid translation using CU recognition increases the understandability for Web documents.

Our future works are as follows. First, increase the ability of the verification using additional rule constraints. Second, compare our mechanism with the alternative practical use of syntactic analyzer as a partial parser. Third, introduce fail softening mechanism as a complement of our pattern-based approach.

REFERENCES

[Bond *et. al.*, 1995] F. Bond, K. Ogura, and T. Kawaoka, Noun Phrase Reference in Japan-to-English Machine Translation, *Proceedings of TMI*, 1995.
[Cho, 1992] Y. Cho, Hangul Trie and Bi-directional Access Method, Technical Report, KAIST (Korean), 1992.
[Choi & Kim, 1996] K. Choi and T. Kim, Current Status and Comparison of Japanese-Korean Machine Translation Systems, *Proceedings of the 2nd Annual Conference on Language Processing* (Japanese), 1996.
[Fredkin, *et. al.*, 1960] E. Fredkin, B. Beranek, and F. Newman, *Trie Memory*, Communications of the ACM 3, 1960.
[Jung *et. al.*, 1997a] H. Jung *et. al.*, Compound Unit Recognition for Efficient English-Korean Translation, *Proceedings of ACH-ALLC*, 1997.
[Jung *et. al.*, 1997b] H. Jung *et. al.*, Multilingual Approach with Compound Unit, *Proceedings of DIALOGUE*,

1997.
[Jung *et. al.*, 1997c] H. Jung *et. al.*, Compound Unit Search on a Trie with Heterogeneous Nodes, *Proceedings of NLPRS*, 1997.
[Katoh & Aizawa, 1995] N. Katoh and T. Aizawa, Machine Translation of Sentences with Fixed Expression, *Proceedings of the 4th Applied Natural Language Processing*, 1995.
[Knuth, 1973] D. Knuth, *The Art of Programming Vol. 3*, Addison-Wesley, 1973.
[Lauer & Dras, 1994] M. Lauer and M. Dras, A Probabilistic Model of Compound Nouns, *Proceedings of the 7th Joint Australian Conference on Artificial Intelligence*, 1994.
[Lee, 1994a] H. Lee, Recognition of Korean-English Bilingual Idioms using Idiom Dispersion Characteristics, Ph.D. Diss., SNU (Korean), 1994.
[Lee, 1994b] S. Lee, Design and Implementation of a Database Index Structure for the Korean Electronic Dictionary, M.S. Diss., KAIST (Korean), 1994.
[Li, 1995] W. Li, Corpus-based Maximal-length Chinese Noun Phrase Extraction, *Proceedings of NLPRS*, 1995.
[Marcus *et. al.*, 1993] M. Marcus, B. Santorini, and M. Marcinkiewicz, *Building a Large Annotated Corpus of English: The Penn Treebank*, Computational Linguistics 19, 1993.
[Schenk, 1986] A. Schenk, Idiom in the Rosetta Machine Translation, *Proceedings of International Conference on Computational Linguistics*, 1986.
[Yoon, 1994] S. Yoon, Efficient Parser to Find Bilingual Idiomatic Expressions for English-Korean Machine Translation, *Proceedings of International Conference on Computer Processing of Oriental languages*, 1994.
[Yosiyuki *et. al.*, 1994] K. Yosiyuki, T. Takenobu, and T. Hozumi, Analysis of Japanese Compound Nouns using Collocational Information, *Proceedings of ICCPOL*, 1994.