Forest Driven Dependency Analysis Enhanced by Japanese Clause Structure Estimation

Satoshi Kamatani, Kentaro Furihata, and Tetsuro Chino

Corporate Research & Development Center, Toshiba Corporation
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki, 212-8582, Japan
e-mail: satoshi.kamatani@toshiba.co.jp

Abstract. For applications of the natural language processing, such as spoken translation, it is the most important characteristic to accept various inputs. We analyze a speech input as a sequence of speech fragment and achieve high acceptability. But it is difficult to determine whether each fragment/unit should be interpreted separately or not. This sort of problem is shared by several applications. We regard a clause and dependency structures as one of the clue to solve this type of problem. Because, even in a spoken language domain, a clause represents a syntactic unit and a dependency represents a semantic unit. In this paper, we describe the dependency analysis method on the syntax forest structure enhanced by Japanese clause structure estimation. It also contributes to improve on translation in manual evaluation from 77.8% up to 78.8%.

1 Introduction

It is the most important characteristic for applications of the natural language processing, such as a spoken language translation (SLT), to accept various inputs. Its capability is directly reflected on availability. We design a SLT system based on this concept. Our SLT system can be divided into two parts. The one is a robust spoken language analysis part, and the other is a rule based machine translation (RBMT) part. The RBMT technology has been successful in commercial business with its ease of maintenance of translation quality and coverage. In contract, it rest on reliable analysis methods. So we proposed a robust parsing method for spoken language [5]. In our proposal, we treat input-utterances as a sequence of fragmental phrases to achieve high acceptability. Then our robust grammar and parser unifies fragmental phrases in several candidates of coherent syntactic units, and enumerates all possible combinations as interpretations. Each interpretation is tested according to syntactic/semantic preferences and the optimum interpretation is to be translated.

But it is difficult to determine whether each fragment should be treated separately or not. This sort of problem is common to all spoken language applications. One key technique is that how to associate each fragment with a suitable unit for a practical application. We regard a clause and dependency structure as one of a clue, since a clause and its dependency represents a syntactic/semantic unit even for a spoken language. In this paper, we propose new analysis method that estimates clause structures and evaluates validities combined with dependency preferences. Our method is designed to work on such a syntax forest like structure. It allows to evaluate all candidates efficiently and to choose totally optimum one.

2 The forest driven spoken language translation

2.1 Overview of our translation process

Figure 1 shows the outline of our SLT system based on the "forest driven spoken language analysis" method [1, 5]. It works based on a packed shared forest structure (syntax forest) generated by extended

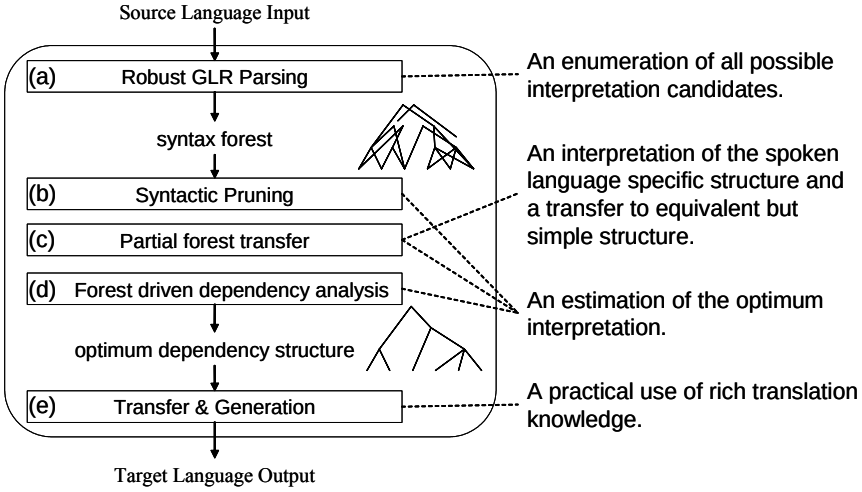

Fig. 1. The processing flow of our SLT system

(1) *C* → *VP*
(2) *NP* → *N CM*
(3) NP_{weak} → *N*
(4) *VP* → *NP V*
(5) *VP* → NP_{weak} *VP*
(6) *VP* → *V*
(7) C_{weak} → NP_{weak}
(8) C_{weak} → *NP*
(9) *SF* → C_{weak}
(10) *SF* → *C*
(11) *UT* → *SF UT*
(12) *UT* → *SF*

UT ··· Utterance
SF ··· Speech Fragment
C ··· Clause
NP ··· Noun Phrase
CM ··· Case Maker
VP ··· Verb Phrase
N ··· Noun
V ··· Verb

Fig. 2. An example of our grammar

Tomita's GLR parser [11]. Here, we notice that we modified the methodology of generating a syntax forest, relative to original one. It is to determine a relation of modification at each vertex. The syntax forest generated by our modified method has following specifications.

- Sub-tree sharing.
  If two or more trees have a common structure, then the sub-tree is represented only once in the parse forest.
- Local ambiguity packing.
  If top vertices satisfy all of following conditions, then they are unified into one vertex.
  1) They dominate a common span of an input sentence
  2) They have a common grammatical category
  3) The head word in the span dominated by them is same

At the step (a) in Figure 1, the parser works with our original grammar. This grammar is designed to capture Japanese clauses and fragmental structures on the basis of following two assumptions;

1. One utterance often consists of a sequence of fragmental phrases.
2. Once some fragments come together as a clause, its internal structure is quite similar to a structure of written language.

Japanese clause consists of at most one subject and one predicate. So, it is comparatively easy to handle. The other characteristic of our grammar is that a grammatically ill-formed part is also accepted as a sub-structure with a special marker "weak". Figure 2 shows a simple example of our grammar. Both rule (2) and rule (3) build a noun phrase, but rule (3) has a "weak" maker. That means, the structure constructed

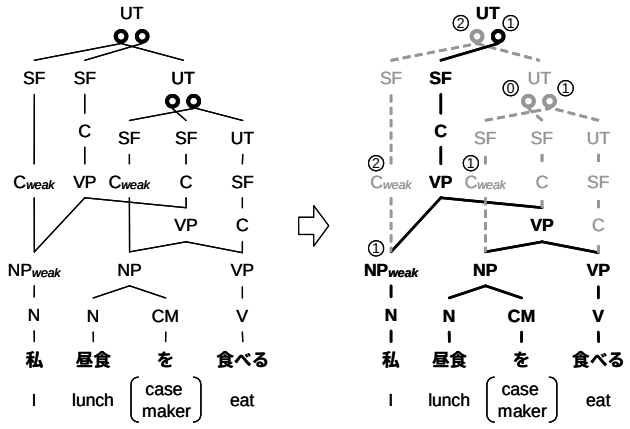

Fig. 3. A sample of the syntactic pruning

by rule (3) is worse than the structure by rule (2). Namely, it represents that a case-marker-omitted structure is syntactically ill-formed.

The step (b) to (d) works to find an optimum interpretation. At the step (b), it estimates max ill-formedness by counting "weak" makers appeared in each sub-structures, and extracts only the parts with least "weak" makers (Fig.3). After that, the partial forest transfer (PFT) method [1] transfers spoken language specific structures to common structures with written one (step (c)). It achieves not only bridging gaps between spoken and written languages but also reducing ambiguity in parses. At the final analysis step (d), "forest driven dependency analysis", we evaluate validities of dependencies in a forest, and choose the optimum structure based on semantic preference. This step is the target of improvements of this paper. So, we will explain the details in the next section.

Once the system chooses the optimum interpretation, at the step (e), we can use the traditional transfer method. If our analysis procedure can not solve all ambiguities, a processed syntactic forest has two or more remained candidates. In such a case, we choose one syntactic tree based on the order of grammar rule.

Our translation method allows utilizing the rich conventional translation knowledge developed for a commercial rule-based machine translation. It ensures high accuracy and broad translation coverage for our system.

2.2 Forest driven dependency analysis

To evaluate the dependency along a structure of a syntax tree is one solution to compare all possible dependency candidates. But, this sort of traditional method usually premises to apply one correct syntactic structure. However our analysis target is a syntax forest that contains huge number of syntax trees. There is no wonder to say that the evaluation cost is expensive. To avoid this problem, we introduce a new analysis method to evaluate directly on a syntax forest [5]. Fortunately, all the possible interpretations are packed and shared as one syntax forest. This feature also means all possible dependencies are packed and shared efficiently. Using this nature, we evaluate all likelihood of dependencies in a forest, and prune semantic ill structures as follows.

1. Estimate the optimum dependency sets of all parse in the forest.
   (a) Put dependency likelihood on all vertices with a modification.
   (b) Aggregate the possible maximum likelihood for each sub-structure by bottom up manner.

2. Extract semantically preferred forest.
   (a) Select vertices with the highest likelihood in each packed sub forest as preferred vertices stepwise top-down manner.
   (b) Extract a well-formed structure consists of only preferred vertices.

Here, we calculate dependency likelihood based on co-occurrence probabilities $p\langle h, rel, m\rangle$. Where h is a head word, m is a modifier word and rel is relationship between h and m. We consider 9 Japanese case markers {ga, wo, ni, de, kara, yori, he, made, no} as a relationship "rel". This co-occurrence probability is leaned from newspaper articles for 7 years by the EM-algorithm based method proposed by Torisawa[12]. But each probability is so small and it has no sense to distinguish a small difference. Therefore, we round a co-occurrence probability down to integer values E_p.

$$E_p(\langle h, rel, m\rangle) = INT(\log_{10}(p\langle h, rel, m\rangle)) \quad (\mathbf{1})$$

where INT is the function to round down to integer. On the other hand, the co-occurrences that denote extremely small probabilities must not be a big clue to the structure decisions. But they are more significant than zero-probability one. So, we define dependency likelihood S_d with the average (1b) of E_p for all triples;

$$S_d(\langle h, rel, m\rangle) = \frac{1}{1+\exp(E_p(\langle h, rel, m\rangle) - \mu_b)} \quad (\mathbf{2})$$

3 Dependency analysis with clause structure estimation

Clause candidates are derived with our grammar, but they are not syntactically discriminated to ensure robustness. And each dependency evaluation is mainly led by a local preference. They are disadvantage for dependency analysis. Besides, it is not necessarily good to determine strictly. In this section, we propose new forest dependency analysis enhanced by estimated clause candidates.

3.1 Detection of clause and its type

Especially in Japanese, it is comparatively easy to estimate the clause boundary and its type by referring local morphological patterns [10]. And Japanese clause types and their characteristics on dependency have been well-studied. Minami [8] distinguished types of dependent clauses according to differences of clause boundaries, and associated each clause type with a degree of independence.

We also use only a sequence of morphemes as clues in the clause estimation. Then, in the forest dependency analysis, we utilize these estimated candidates of a clause to evaluate validity of dependency. Concretely, we define rules to detect a clause candidate and determine its syntactic and semantic characteristics. One rule consists of three parts, a feature name, a morpheme pattern, and a dependency controls. Now, we explain the detail of this clause detection rule by using Table 1. A feature name is to be assigned a input morpheme to distinguish clause candidates. In Table 1, the rule (1) has a name "cp_ga". A morpheme pattern is consisted of a sequence of word stems and POS tags and used to

Table 1. Samples of clause detection rules

Feature name	Pattern	Penalty
cp_ga	-1=*&0.pos="conjunctive particle"&0.surface="が"	cm_ga;-0.9
cm_ga	-1.pos="noun"&0.pos="case maker"&0.surface="が"	cm_ga;-1.0

detect a candidate of a clause. It is described in the extended regular expression. The pattern expressed in the rule(1) says that POS of noticed morpheme is conjunctive particle, a stem is “が”, and an arbitrary morpheme exists at the previous of the noticed morpheme. A dependency control consists of a condition to fire and a penalty when the proceeding condition is fired. The rule (1) is fired if it conflicts with feature cm ga and assigns penalty points -0.9.

Each rule is also defined whether they are active or not in modifier and modified vertices. This activation information depends on only types of the clause. For a following experiment, we used types of clauses listed in [8], and developed 51 rules by hand. Here, we have to notice that penalties in each rule were experimentally adjusted.

3.2 The enhanced dependency analysis procedure

We describe a detail of dependency analysis method enhanced by detected clause candidates. The pruning procedure works along the extended forest driven dependency analysis and prunes semantically ill structures as follows. Now, we use a syntax forest described partially in Figure 4 as a conceptual diagram. In this figure, a head word inheritance is represented as a arc. The bold arc denotes a head arc. The other thin arc denotes to be a non-head arc and a modification.

1. Estimate clause candidates with detection rules, and assign features.
2. Estimate the optimum dependency sets of all parses in the forest.
   - A) Put a likelihood calculated by the equation (1) for all vertices.
   - B) For each modified vertex, collect features with following conditions in a span of inputted morphemes dominated by the vertex.
     - The features are assigned for the head morpheme of the vertex.
     - The features are defined that they are active in a modified vertex.
   - C) For each modifier vertex, assemble features with following conditions in a span of inputted morphemes dominated by the vertex.
     - The features are assigned for the head morpheme of the vertex.
     - The features are assigned for the morpheme appeared after the head morpheme of the vertex.
     - The features are assigned for the morpheme appeared before the head morpheme of the vertex and defined that they are active in a modifier vertex.
   - D) For each feature assigned for modified vertices, evaluate conditions in detection rules and find the first satisfied condition.
   - E) If the satisfied condition is found, assign the penalty defined in the rule to the vertex and note the vertex where the morpheme which assigned the feature are unified to the head morpheme of vertex.
   - F) At each vertex, sum up penalty and likelihood as the score.
   - G) Aggregate the possible maximum score for each sub-structure by bottom up manner.
3. Extract semantically preferred forest.
   - A) Select vertices with the highest likelihood in each packed sub forest as preferred vertices stepwise top-down manner from root of the forest. If the punished vertex is noted, the vertex is pruned from the preferred structure.
   - B) Extract a well-formed structure which consists of only preferred vertices.

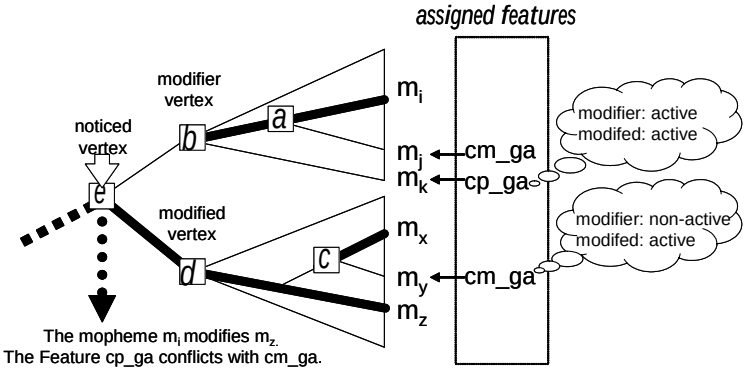

Fig. 4. Forest driven dependency analysis with a clause estimation

This procedure allows us to integrate local preference of (Japanese) clause structure estimated from a fragmental and sequential input, syntactic preference derived from grammar rules, and semantic preference calculated on co-occurrence probabilities. Therefore, we can effectively acquire totally preferable structures.

4 Evaluations

In our experiments, we used 702 utterances extracted from a parallel corpus made by ourselves. This parallel corpus is a collection of Japanese-English translation pairs in a travel domain. These extracted utterances are open data for our SLT system, and have about 13.4 characters and about 7.2 words on average.

4.1 Effectiveness on the dependency analysis

To evaluate the validity of the pruning based on the method proposed in section 3.2, we used 200 utterances randomly extracted from the 702 utterances. These 200 utterances have about 17.3 characters and about 9.4 words on average. Correct dependency relations for each utterance were assigned by hand.

Our grammar is built with high redundant rules to accept several input phenomena. Accordingly, the grammar acquires robustness, while it generates different structures in a same interpretation. Therefore, what we have to evaluate are that 1) the pruning operation prunes ill-structures as many as possible, and 2) after the pruning operation, a syntax forests keeps the correct dependencies. At this point of view, we define an original measure “Correct Dependency Ratio”;

$$CDR = \rho \cdot \frac{\sum_{t \in F} \frac{D_m}{D_c}}{N_F} \qquad (3)$$

where ρ takes “1” when the all correct dependency are enclosed in a syntax forest, otherwise it takes “0”, F is a syntax forest, t is a syntax tree in F, N_F is a number of syntax trees in the F, D_c is a number of correct dependencies, and D_m is a number of dependencies in t that matches to correct dependencies.

Table 2. Reduction of the parse number and Improvement of CDR.

	Parse Num.		CDR	
Clause estimation	off	on	off	on
Accepted forest	1830.1		0.710	
Syntactic Preferred forest	5.58		0.833	
Semantic Preferred forest	3.27	2.80	0.851	0.875

Table 3. Translation accuracy

System	Accuracy	NIST
Commercial MT	56.7%	3.24
without clause estimation	77.8%	3.44
with clause estimation	78.8%	3.64

Table 2 shows the average number of syntax trees in syntax forests at each step. In this table, "Accepted Forest" is for the forest after step (a) in Figure 1, "Syntactic Preferred Forest" is one for after step (b), and "Semantic Preferred Forest" is one for after step (d). And Table 2 also shows the average of CDR for each step. To discuss the significance statistically, we assessed t-test (level of significance R = .05), and verified a significant difference. Those results lead that our method efficiently eliminates ill-syntactical structure and ill-dependency, without losing a valid interpretation.

4.2 Effectiveness on the translation

We evaluated all of 702 utterances translated by three systems, the forest driven SLT system 1) without the estimation of clause structure, 2) with the estimation of clause structure, and 3) a commercial translation system developed for written languages. The third system is the source of transfer rules used at the step (e) in Figure 1. Each translated utterance is examined by three tester, and we determine whether a translation is correct or not by majority.

Table 3 shows the evaluation of translations. In this table, the NIST [2] scores are also shown for reference. It shows an improved accuracy rate for translation treated with the clause estimation. Using the clause estimation, translation results of 21 utterances are changed, of which 9 utterances are newly judged correct translation, and of which 10 utterances are improved quality of translations. Worsened 2 utterances are caused by lack of transfer rules. Therefore, the analysis process enhanced by our method contributes to improvements.

5 Related works

We can regard the sentence splitting methods as another way to handle fragmental inputs. If the certain splitting points are found, each split part can be processed and translated better. In the domain of a written language, it is often solved by splitting an inputted sentence, such as Roh [9] and Zhang [13]. They focused on written sentences that are comparatively so long and well-ordered. So, the splitting point could be strictly determined. In the spoken language domain, it's often the case that it addresses to determine the end of an utterance, such as Lavie [6] and Takanashi [7]. They are useful to cut out a segment to parse, but they are deterministic and do not supply preference of a relation in each segment. On the contrary, Furuse [3] proposed the method splitting into well balanced translation units based on a

semantic distance calculation. It successfully splits one utterance into stored patterns, but it is necessary to prepare rich translation patterns.

On the other hand, the other way to choose the optimum structure from a forest structure is to use a probabilistic GLR (PGLR) parsing method [4]. But, it is difficult to prepare corpora enough to learn statistical models about specific phenomena of spoken language. And statistic based methods also have a difficulty of a maintenance by hands.

Our method estimates clause structures with the robust grammar and surface patterns of an utterance, without strict assumption. And it unifies preferences of clause structures and dependencies. That ensures to incorporate a statistical method with a human controllable method. Then the system can choose the optimum interpretations from all possible candidates.

6 Conclusion and future works

We proposed the forest driven dependency analysis enhanced by the estimation of Japanese clause structures. It enables to effectively integrate local preference of clause structures, syntactic preference, and semantic preference. It helps us to choose the totally preferable structure from a syntax forest efficiently. The result was our forest driven SLT system successfully improved quality of translations.

For the further improvement, we have to study on combination with machine learning. Using a machine learning method, we plan to find a clause candidate and syntactic/semantic behaviors. Especially, it is necessary to refine penalties which are experimentally assigned by hand. In addition, while we focused Japanese clause structures in this paper, it is possible to use key idea of our method for other languages by using chunk.

References

1. Tetsuro Chino and Satoshi Kamatani. Partial forest transfer for spoken language translation. In *Proceedings of RANLP'05*, pages 157-161, 2005.
2. G. Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of ARPA Workshop on Human Language Technology*, 2002.
3. Osamu Furuse, Setsuo Yamada, and Kazuhide Yamamoto. Splitting long or illformed input for robust spoken-language translation. In *Proceedings of COLINGACL'98*, pages 421-427, 1998.
4. Kentaro Inui, Virach Sornlertlamvanich, Hozumi Tanaka, and Takenobu Tokunaga. New formalization of probabilistic glr parsing. In International Workshop on Parsing Technologies, pages 123.134, 1997.
5. Satoshi Kamatani, Tetsuro Chino, and Kazuhiro Kimura. Syntax forest based syntactic and dependency analysis towards robus spoken language processing. In *Proceedings of PACLING'05,* pages 192-199, 2005.
6. Alon Lavie, Donna Gates, Noah Coccaro, and Lori Levin. Input segmentation of spontaneous speech in janus: A speech-to-speech translation system. In *Proceedings of 12th ECAI*, pages 86-99, 1996.
7. Takashi Masuoka and Yukinori Takubo. Kiso Nihongo Bunpou . Kaitei-ban .Kuroshio-Shuppan, 1992. (in Japanese).
8. Fujio Minami. Gendai Nihongo-no Kouzou. Taishukan-Shoten, 1974. (in Japanese).
9. Yoon-Hyung Roh, Young-Ae Seo, Ki-Young Lee, and Sung-Kwon Choi. Long sentence partitioning using structure analysis for machine translation. *In Proceedings of NLPRS2001*, pages 646-652, 2001.
10. Katsuya Takanashi, Takehiko Maruyama, Kiyotaka Uchimoto, and Hitoshi Isahara. Identification of "sentences" in spontaneous Japanese detection and modification of clause boundaries. In *Proceedings of ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, pages 182-186, 2003.
11. Masaru Tomita. Generalized LR Parsing. Kluwer Academic Publishers, Norwel, Massachusetts, 1991.
12. Kentaro Torisawa. An supervised method for canonicalization of Japanese postpositions. In *Proceedings of NLPRS2001*, pages 211-218, 2001.
13. Yujie Zhang and Kazuhiko Ozeki. The application of decision trees to segmentation of long Japanese sentences. *Natural Language Processing*, 7(1):13-30, 2000. (in Japanese).