Utility for Communicability
by Profit and Cost of Agreement

Ryuichi Matoba, Makoto Nakamura, and Satoshi Tojo

School of Information Science,
Japan Advanced Institute of Science and Technology,
1–1, Asahidai, Nomi, Ishikawa, 923–1292, Japan
{r-matoba, mnakamur, tojo}@jaist.ac.jp

Abstract. The inflection of words based on agreement, such as number, gender and case, is considered to contribute to clarify the dependency between words in a sentence. Our purpose in this study is to investigate the efficiency of word inflections with HPSG (Head–driven Phrase Structure Grammar), which is able to deal with these features directly. Using a notion of utility, we measure the efficiency of a grammar in terms of the balance between the number of semantic structures of a sentence, and the cost of agreement according to the number of unification processes. In our experiments, we showed how these were balanced in two different corpora. One, WSJ (Wall Street Journal), includes long and complicated sentences, while the other corpus, ATIS (Air Travel Information System) does shorter colloquial sentences. In the both corpora, agreement is surely important to reduce ambiguity. However, the importance of agreement in the ATIS corpus became salient as personal pronouns were so often employed in it, compared with the WSJ corpus.

1   Introduction

Grammatical rules of human language enable us to generate an infinite range of expressions. Because a long sentence may contain ambiguities in its meaning, language is equipped with devices to indicate dependency. Among them, we consider the function of agreement.

Types of agreement such as number, gender, and case change word forms by inflection or agglutination of prefix/suffix. Inflection is a grammatical affix that attaches to a word to mark it as a particular part of speech [3], in English, for example, the use of -*ed* to make the word *show* into the past tense form *showed*, and the use of -*s* to make the word *actor* into the plural *actors*. Agglutination is to combine words with sets of stable affixes to produce complicated phrases like *judgemental* or *helplessness* in English. In technical terms, the case change of nouns or adjectives is called declension, and words are classified into declinable ones and other indeclinable ones, while verb inflection is called conjugation [1]. Among we said above, those words which are often used in daily expressions, such as, *I, my, me, mine*, or irregular verbs tend to totally change their forms. However, these irregular inflections are an obvious barrier for language learners such as toddlers and foreigners.

These types of agreement are considered to assist to identify or reidentify referents [2]. The languages with rich inflectional systems often allow more freedom in word order, while those which only have few inflectional systems such as modern English requires strict word order instead [3,6]. This relationship between word order and agreement is one of the sign to represent characteristics of languages. Because these grammatical features are considered to be diachronically developed, studying languages through the aspect of agreement and word order may contribute to the field of language evolution.

Kirby's model [8], claimed that human beings had a skill to develop and create grammars spontaneously for compositionality and recursion. Considering these two functions, we presume that we can generate an infinite variety of expressions. However, his model seems rather to have neglected the aspect of understanding. Such a free proliferation of language may decrease the communicability, if we can generate long and complicated sentences, then they inevitably include ambiguities. Thus, we contend that a grammatical feature which decreases ambiguity such as agreement should also contribute to developing spontaneity. Our work is based also on Jäger [7], who has used the notion of *utility* to describe the efficiency of language generation. We will utilize this notion, and define our own utility function later. In our study, we bring HPSG (Head–driven Phrase Structure Grammar) which is indebted to a wide range of research traditions in syntax, like categorial grammar, generalized phrase structure grammar, lexical–functional grammar etc. Compared with other grammars dealing with phenomena of language evolution [4,12], HPSG is simple enough to handle our experiment. Because in HPSG every part of speech, category, and partial tree is represented by feature structures or DAG's (directed acyclic graphs), we can embed the restrictions on agreement in grammar formalisms.

In this study, we scramble the word order of sentences artificially, and measure how they become ambiguous. For this purpose, (i) we define a utility function, the value of which is higher if the meaning is less ambiguous with less effort. (ii) Next, we show the difference of utility values between inflectional and non-inflectional languages. We expect that inflectional language will have less ambiguity, though it will cost more in the parsing process. We show the value of the utility against the ratio of randomization of word order. (iii) Then, we compare two sample corpora. One is ATIS which consists of rather simple short sentences. The other is WSJ, which includes many complicated sentences. We hypothesize that in ATIS the non-inflectional language will suffice for communication and show high values in the utility, while in WSJ the inflectional one will show high values.

This paper is organized as follows: in Section 2 we explain the mechanism of HPSG and a programming language which we use to build our system. Section 3 presents the details of our experimental model, as well as the explanation of the utility function. We show our experimental results in Section 4, and Section 5 summarizes this study and outlines the further work.

2   HPSG and Parser

2.1   Briefing HPSG

Feature structure, Type, and Head. In HPSG, each word and phrase is not a single symbol but a set of feature–value pairs called a *feature structure.* A feature structure is a directed graph, in which all the nodes and edges have associated names. The name associated with a node is called *type* and the name associated with an edge is called *feature.* The types of edges or attributes that can be associated with a node are determined uniquely by the type. In case the feature takes multiple candidates as its value, it is represented by a list '$\langle \ldots \rangle$'.

In Fig. 1, we illustrate a parsed tree of 'A man walks.' Note that each node of the tree is a feature structure; because the value of feature NUM is fixed as '3sg' (third person, singular), feature structures with different NUM values cannot be accepted to form a tree.

We metaphorically call the upper node a *mother* and the lower one a *daughter* in a tree structure. In a verb phrase or a noun phrase, we can find the prime daughter who mainly decided the features of her mother, that is, a verb and a noun, called *head.*[1] In Fig. 1, the head of the phrase 'A man' is a noun 'man.'

In HPSG, various categories are classified into types. Each category, that is a feature structure, owns a type and it is usually placed at the top of the whole structure, headed by '~' (tilde).[2] The upper type inherits the features of all the lower types, and thus, all the types form a *type-hierarchy.* We use '$\sqsubseteq$' for the *subsumption* relation in types. The feature structure of type a of $a \sqsubseteq b$ is a subset of the structure of type b. The bottom type ($\bot$) is the most general type with no features.

Heads are often represented by a bar '⎯' over the type labels. Because the type of 'man' is N, that of 'a man' becomes $\overline{N}$. Similarly, the head of a verb phrase $\overline{V}$ is a verb V, and because the whole sentence can be considered to be formed by a verb phrase, the type of the sentence becomes $\overline{\overline{V}}$. The process in which a head is combined with other daughters into a mother is called a *subcategorization*, such that 'man' (N) takes an article 'A' to $\overline{N}$, or 'walks' ($\overline{V}$) takes 'A man' ($\overline{N}$). As in Fig. 1, a type requires other types specified in SUBCAT feature to be a mother structure.

ID-schemata and Principles. The grammar rules of HPSG consist of *ID-schemata* and principles. An ID-schema corresponds to a generation rule of Context–Free Grammar (CFG), viz., the left-hand side of '→' is a mother category and the right-hand side is daughter categories. The feature HEAD contains the structures of the head feature and DTRS (daughters) contains the structures

---

[1] Some linguists define that the head of a noun phrase is a determiner because without it a noun cannot be a phrase, but the discussion is out of the scope of this paper.

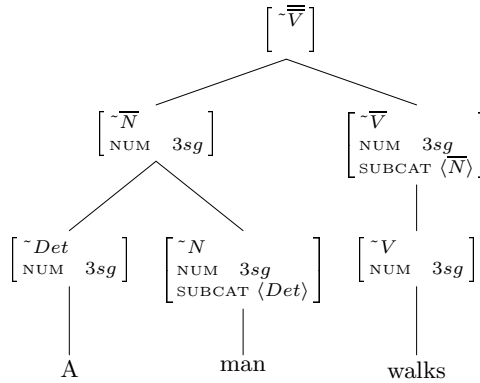[2] In this paper, we may omit type labels unless they are necessary.

Fig. 1. A tree structure of a natural language sentence

of HEAD-DTR and COMP-DTR (complement daughter). Consecutive feature names delimited by '|' access the value of the last feature.[3]

A *pointer* indexed by '[i]' points to another structure headed by the same label, and denotes that the labeled structure is shared. The operation which merges multiple feature structures into one, without losing consistency, is called *unification.* This is utilized when looking up lexicons and grammar rule applications.

Principles are the constraints which all the feature structures must satisfy a priori. For example, *Head Feature Principle*, shown in Eqn (1), declares that both mother HEAD features and those of daughters must be common.

$$
\begin{bmatrix}
\text{||HEAD} \quad \boxed{1} \\
\text{DTRS|HEAD-DTR} \quad \begin{bmatrix} \text{||HEAD} \quad \boxed{1} \end{bmatrix}
\end{bmatrix}
\tag{1}
$$

Subcategorization Principle, shown in Eqn (2), limits the mother SUBCAT feature to the list of all the daughter SUBCAT features minus those which are already subcategorized.

$$
\begin{bmatrix}
\text{||SUBCAT} \quad \langle \; \boxed{2} \; \rangle \\
\text{DTRS} \begin{bmatrix}
\text{HEAD-DTR} \begin{bmatrix} \text{||SUBCAT} \; \langle \; \boxed{1} \, , \; \boxed{2} \; \rangle \end{bmatrix} \\
\text{COMP-DTRS} \; \boxed{1}
\end{bmatrix}
\end{bmatrix}
\tag{2}
$$

The parsing process of HPSG is to acquire the mother feature structure, filling out its initially vacant HEAD-DTR and COMP-DTRS features, taking two other structures, with ID-schemata and principles. Although the order of this composition is not specified, the possible combinations are limited by the given initial set of structures.

---

[3] Hereafter, we abbreviate the description of consecutive feature names. '|' denotes that several feature names which are located the left side of two vertical bars are omitted.

2.2 LiLFeS and UP Parser

LiLFeS[4] is a programming language that processes linguistic formalisms, developed at University of Tokyo [9]. It is similar to Prolog in that its syntax includes disjunction, negation, and cut. The library of the language has been well developed, and we can utilize such functions as copying of feature structures, multi-dimensional arrays, regular expressions, and so on. Thus, LiLFeS is well suited to implement HPSG.

The UP parser which is written in LiLFeS is included in the MAYZ toolkit[5]. It analyzes a sentence according to a unification–based grammar, e.g., HPSG. Users can give the target sentence as a string or a word lattice to the parser.

Using these two tools, we implemented our HPSG parser which enables us to derive Predicate-Argument Structures (PASs) not only from ordinary English sentences but also from artificial ones with scrambled word order.

3 Utility Investigation Model

3.1 Word Order Variation

In this section, we introduce the methods to represent scrambling in HPSG [5]. In Japanese, for example, beside the standard subject–object–verb (SOV) order, the object–subject–verb (OSV) order is admissible as below, both of which give the same meaning as "Ken loves Naomi."

Ken–ga (NOM) Naomi–wo (ACC) aisiteiru (verb).
Naomi–wo (ACC) Ken–ga (NOM) aisiteiru (verb).

The syntactical structures of the above two sentences are composed using SLASH feature. Briefly, if a category contains the SLASH feature it dominates a gap (missing constituent). To help readers identify the location of the gaps, we mark them with an underlined space in the following example sentences.

What did you say they handed _ to the baby?
The presents that it annoys me that the children discover _ ...

The gap in the phrase "hand to the baby" from which an NP is absent is represented as [SLASH <NP>], as in Fig. 2 [11].

In this paper, because we basically deal with English grammar, we adopt this SLASH feature to represent word migration. Using the SLASH feature, a verb, "love," is represented in Fig. 3.

The languages of the Latin family such as Italian and Spanish apply the rule of the left-hand side of Fig. 3 when an object of a verb is a pronoun. The Germanic languages whose word order is called SOV-V2 such as German and Netherlandic
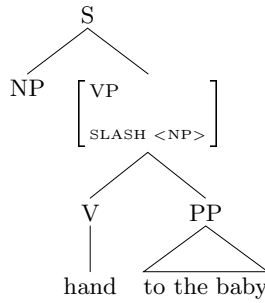
---

[4] http://www-tsujii.is.s.u-tokyo.ac.jp/lilfes/
[5] http://www-tsujii.is.s.u-tokyo.ac.jp/mayz/

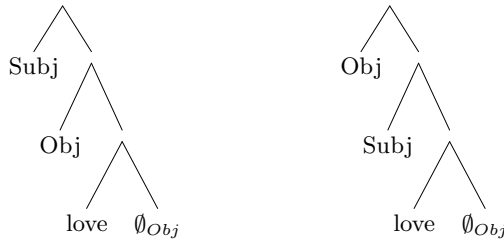**Fig. 2.** Example of handing SLASH feature in HPSG

Fig. 3. Lexical item of "love" for the scrambled expression

apply this rule. The rule of the right-hand side of Fig. 3, long distance dependency, is applied to question forms and relative clauses of English. According to these migrating rules, a sentence "Subj *loves* Obj" may change to "Obj Subj *loves*" or "Subj Obj *loves*", but may not change to "*loves* Subj Obj" or "*loves* Obj Subj'.'

3.2 Parsing Process of Scrambled Sentences

We explain the characteristics of our parser, shown in Fig. 4, which is an example of a parsed tree after scrambling of "she loves me" in HPSG. In this figure, *nom, acc*, HSS, HCS, HFS are the abbreviations of nominative (Subject), accusative (Object), head–subject schema, head–complement schema, and head–filler schema, respectively. These schemata are explained as follows:

Head-complement schema. This schema is for a verb phrase to take a complement. Only those words, the category of which has been specified in SUBCAT, can be unified to the verb phrase.

Head-subject schema. This schema is for a verb phrase to take a subject. The subject candidate can be applied to the schema only when the verb phrase has satisfied all the complements.

Head-filler schema. This schema is for a relative clause, and wh–question. Only a word which is categorized as SLASH can be unified as a complement.

With the rule of the right-hand side of Fig. 3, the sentence is parsed as in Fig. 4(a) by the SLASH feature. The verb "loves" in which the first argument
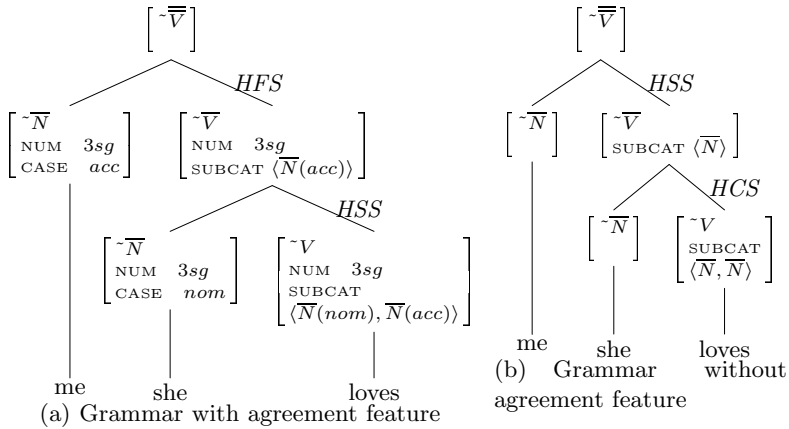
Fig. 4. Tree structures of a scrambled sentence "me she loves"

of the SUBCAT is a nominative noun requires the noun "she" whose case is nominative by HSS. Because the SUBCAT of the verb phrase "she loves" is an accusative noun, the parser applies HFS to the accusative noun "me".

In Fig. 4(b) shows the example of ignoring agreement. The information of the SUBCAT of the verb "loves" lacks the information on case. Therefore, the parser cannot figure out whether the noun "she" is the subject or the object of the verb. At this time, Using HCS, the verb phrase "she loves" is constructed. The SUBCAT of the verb phrase "she loves" is the noun, and the only remaining noun is "me"; this verb phrase does not have a subject yet, thus with HSS this verb phrase subcategorizes a noun "me".

The output of this parser is a Predicate–Argument Structure (PAS), that is one of the characteristics of HPSG. Figure 5 shows the outputs of parsing dependent upon the agreement features. "Pred (argument1, argument2)" means that "argument1 Pred argument2". For the parsing result with the agreement features in Fig. 4(a), we can figure out what this sentence means without any ambiguity, though the sentence is scrambled. However, the result of Fig. 4(b), in which the grammar does not consider agreement features such as CASE and NUM, shows that we cannot figure out the meaning either "me loves she" (meaning "I love her") or "she loves me."

3.3   Experimented Procedure

In this section, we explain the procedure of our experiment (see Fig. 6). Due to defects of the grammar set of our parser, not all the sentences in the corpora can be parsed. Therefore, first, we extract only those sentences which can be parsed. Next, we scramble the word order of the parsed sentences, and embed them into each original corpus at the rate of 0 % to 100 %. Then, we parse them with our HPSG parser which tolerates scrambled sentences. Also using a function of LiLFeS which can ignore arbitrary features, we parse the sentences disregarding agreement.
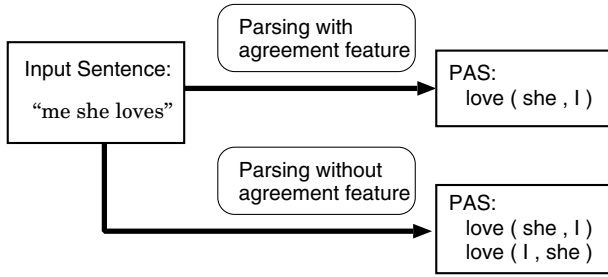
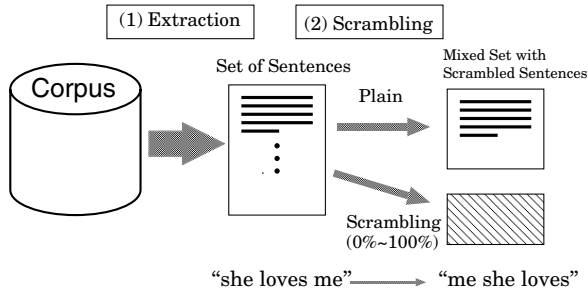**Fig. 5.** Outputs of parsing depend on agreement

Fig. 6. Procedure of experiment

Finally, using these outputs, we calculate a value of the utility to evaluate grammars with regard to the balance between word order and the cost of unification.

3.4 Utility

In our model, based on the utility formula of Jäger's research [7], we propose a utility function for a sentence, U, such that:

$$U = \frac{N}{|PAS|} - w \cdot C(unif), \tag{3}$$

where $|PAS|$, N, $C(unif)$, and w are the number of predicate argument structures extracted from the sentence, the word length of the sentence, the number of unifications for the agreement feature, and the weight for $C(unif)$, respectively.

As the number of PASs represents ambiguity, the first term of (3) indicates profit. The less ambiguous the sentence is, the higher the value of the first term is. The second term of (3) indicates cost of unification. In Fig. 4(a), for example, the number of unification of agreement is four times, while the number of unification of agreement in Fig. 4(b) is zero. The low value of this second term means that the sentence is parsed more efficiently. The weight w is a positive coefficient by which the priority of the second term is assessed.

4   Experiment and Result

In this section, we show our experimental results, based on the model in the previous section. The purpose of this experiment is to observe the utility value against the proportion of the scrambled sentences. In our experiment, we set the weight in Eqn (3) to $w = 0$ and 0.1.

We apply our system to two corpora recorded in the Penn Tree Bank; one is the Wall Street Journal (WSJ, hereafter), and the other is the Air Ticket Information Service (ATIS). These corpora contrast with each other; while the WSJ includes many grammatically complicated sentences, the ATIS consists of comparatively short and simple sentences. We extracted 5,856 sentences out of the WSJ corpus. After artificially scrambling them, we have generated 4,390 sentences. In the similar way, we have employed 210 sentences out of the ATIS corpus. From each of these sets of sentences, we randomly sample 10,000 sentences with replacement, and calculate their value of the utility. We have parsed these word-free sentences with four different grammars, considering the requirements of agreement as to person/number and case marking. The semantics of a sentence is shown as a PAS in HPSG. If a sentence is multiply parsed, we obtain multiple PASs and thus the sentence is ambiguous.

We show the results of the WSJ and the ATIS in Fig. 7 and Fig. 8, respectively. Figure 7(a) is the result of the WSJ at $w = 0$, and Fig. 7(b) at $w = 0.1$. The same holds for Fig. 8. The horizontal axis denotes the mixture ratio of scrambled sentences, and the vertical axis is the average of utility. The solid line tagged by *PERNUM&CASE* shows the grammar in agreement both with person/number and case marking, the long dashed line by *CASE* denotes the one regarding only case, the short dashed line by *PERNUM* regarding only person/number, and the dotted line by *NONE* is the grammar disregarding agreement. The chain line tagged by *Difference* denotes the difference of the utility between *PERNUM&CASE* and *NONE*. All these lines include spline interpolation.

As we can see in Fig. 7and Fig. 8, the more the mixture ratio of scrambled sentences increases, the more the utility values decrease. According to Eqn (3), the profit term is inversely proportional to the number of PASs, which implies the degree of ambiguity for a sentence. Because, in most cases, scrambled sentences are more difficult to identify a nominative and an accusative of a verb than normal sentences with strict word order of English, the average number of PASs for the scrambled sentences is greater than that of the normal ones. As a result, the increase of scrambled sentences simply makes the average of utility low.

Here, we focus on Fig. 7(a). The weight in Eqn (3) is set to $w = 0$, which means that language users do not care the cost of agreement. The utility value of *PERNUM&CASE* is the highest over all the mixture ratio of scrambled sentences, while that of *NONE* is the lowest. Because the grammatical features such as agreement with person/number and case marking are considered to contribute to identify referents, the grammar using them reduces the number of PASs. The figure represents the effect of these grammatical features in the order of agreement with person/number and case marking (*PERNUM&CASE*), only person/number (*PERNUM*), and only case marking (*CASE*). Because the
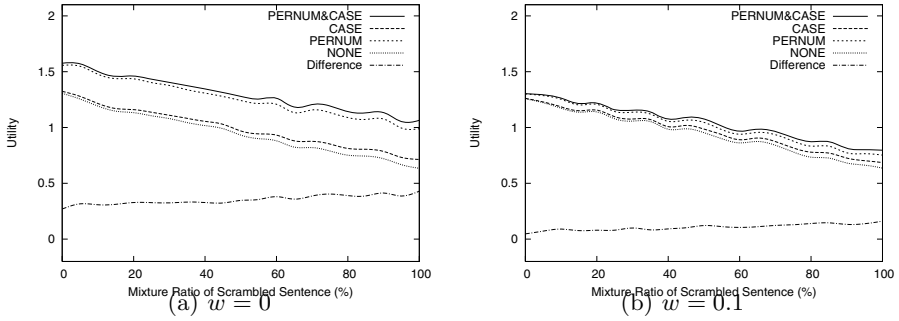
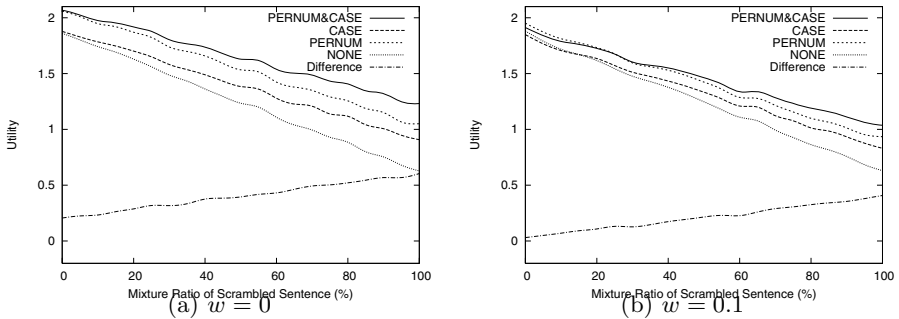**Fig. 7.** Utility and the ratio of scrambled sentences: WSJ

Fig. 8. Utility and the ratio of scrambled sentences: ATIS

frequency of unification processes for agreement with person/number was 14.7 times as much as that of case marking, the grammatical feature for agreement with person/number is considered to be more efficient than that of case marking. We can see that the more the mixture ratio increases, the greater the difference between *PERNUM&CASE* and *NONE* is. Thus, the use of these features for scrambled sentences is more effective in the reduction of ambiguity. It is consistent with the fact that the languages with rich inflectional systems often allow more freedom in word order.

Figure 7(b) shows the result of the WSJ at $w = 0.1$. In this situation, language users are annoyed by taking agreement. Therefore, the utility value of *PERNUM&CASE* becomes close to that of *NONE*, and when $w = 0.2$, it eventually becomes worse than *NONE*. Second language learners may not be accustomed to dealing with agreement, i.e., in their case, the weight of cost becomes large. This means that the grammar disregarding agreement brings them a high value of the utility.

We show the result of the ATIS at $w = 0$ in Fig. 8(a). Although the order of the utility values for each grammatical feature is the same as the WSJ, it is entirely higher than that of the WSJ. Compared with the WSJ as written English sentences, the corpus of the ATIS as colloquial sentences contains many

of shorter sentences. Because long sentences tend to own the vast amount of ambiguous expressions, the average utility value of the WSJ is likely to be less than the ATIS. Despite the long sentences contain more complicated structures in the PASs than shorter ones, we do not consider the complexity of PASs in a sentence in the definition of the utility in Eqn (3). Hence, the word length of a sentence is properly not reflected in the definition of the utility.

Paying attention to the difference of the utility between *PERNUM&CASE* and *NONE*, we can observe the increase rate of the difference against the mixture ratio of scrambled sentences is greater than that of the WSJ. In other words, the grammatical features are more effective in scrambled sentences in the ATIS than that in the WSJ. Because of a heavy usage of personal pronouns in the ATIS corpus, the case marking is more important than in the WSJ. In fact, 28.1 % of the sentences contained personal pronouns in the ATIS while 10.9 % in the WSJ.

We show the result of the ATIS at $w = 0.1$ in Fig. 8(b). When scrambled sentences are rarely contained, the grammar with only person/number, that is *PERNUM*, exceeds the one with both person/number and case marking, that is *PERNUM&CASE*, in terms of the utility values. Similarly, the utility values of the grammar with case is below the one disregarding agreement. This phenomenon comes from the unification cost for case marking. On the other hand, because grammatical feature of case is important to the disambiguation of scrambled sentences, the utility values for *PERNUM&CASE* and *CASE* are still kept high, even if the mixture ratio of scrambled sentences increases.

5   Conclusion and Future Work

In this study, we have investigated the utility of word inflections based on agreement, which can clarify dependency and can reduce the ambiguity. We have measured the ambiguity by (a) the number of probable PASs, i.e., possible semantic structures of a sentence, and (b) the cost of agreement by the number of unification processes. We defined our utility function by (a)−(b) and showed how these were balanced in two different corpora. One was the Wall Street Journal corpus which contains long and complicated sentences, in the other corpus, Air Travel Information Service corpus which contains shorter sentences.

In the experiment, we prepared four types of grammar which are (i) the grammar regarding person/number and case, (ii) only person/number, (iii) only case, and (iv) disregarding all of them.

As a result, the more the mixture ratio of scrambled sentences increases, the more the utility value decreases. Also, we have observed that the more the mixture ratio increases, the greater the difference between the grammar with agreement and the one without it. Because the word order is strict in English, the grammar without agreement has shown higher utility than those with any agreement. For those who are accustomed to dealing with agreement, Grammar (i) showed the highest utility. On the other hand, for those who are not, Grammar (iv) became efficient. The latter case may often happen in the second language acquisition.

Thus far, in many studies concerning language change and language evolution, grammar has only been regarded as an abstract notion, as a virtual parameter set of Universal Grammar. However, in this study, we actually wrote grammar rules to investigate how each rule affects the features of language. We owe the benefits of our experiment greatly to the adoption of HPSG, which is one of the advanced style of grammar formalism, in that we could calculate the profit by the number of PASs in the analysis of SLASH features, and the cost by the number of feature unification. However, in this grammar, there are many other features which we have not dealt with yet. Employing these features, we will be able to analyze practical phenomena more precisely, also modeling in the future.

In our system, understanding of a speaker's utterance in actual situations corresponds to deriving a unique PAS from the sentence. Our present rule–based parser, however, does not disambiguate multiple PASs semantically, i.e., we do not determine a unique PAS from the multiple candidates. For solving this problem, a stochastic parsing method is necessary. If the parser chose one from the candidates, we could calculate a precision of the PASs against the correct answer, which is recorded in the corpora as parsed tree structures. Also, in this study, we have not considered the complexity of PASs in a sentence in the definition of the utility, even if a long sentence contains complicated structures in the PASs than shorter ones. In the near future, we will develop a stochastic system based on Enju[6], which is a stochastic HPSG parser [10], and then redefine a new utility including a precision and a term for complexity of PASs.

References

1. J. Anward. Word Classes / Parts of Speech : Overview. In Keith Brown, editor, *The Encyclopedia of Language and Linguistics*, Vol. 13, pp. 628–629. Elsevier Science Publishers, Amsterdam, NL, second edition, 2006.
2. M. Barlow and C.A. Ferguson, editors. *Agreement in Natural Language: Approaches, Theories, Descriptions*. CSLI, Stanford, CA, 1988.
3. B.J. Blake. *Case*. Cambridge University Press, 1994.
4. T. Briscoe. Grammatical acquisition and linguistic selection. In *Linguistic Evolution through Language Acquisition*. Cambridge University Press, 2002.
5. T. Gunji. *Japanese Phrase Structure Grammar*. Reidel, Dordrecht, 1987.
6. R. Jackendoff. *Foundations of Language: Brain, Meaning, Grammar, Evolution*. Oxford University Press, 2002.
7. G. Jäger. Evolutionary Game Theory and Typology:a Case Study. In *Proceedings of the 14th Amsterdam Colloquium*, pp. 108–117, 2003.
8. S. Kirby. Learning, Bottlenecks and the Evolution of Recursive Syntax. In *Linguistic Evolution through Language Acquisition*. Cambridge University Press, 2002.
9. T. Makino, K. Torisawa, and J. Tsujii. Lilfes–Pactical Programming Language For Typed Feature Structures. In *Natural Language Pacific Rim Symposium '97*, 1997.

---

[6] http://www-tsujii.is.s.u-tokyo.ac.jp/enju/

10. Y. Miyao and J. Tsujii. Probabilistic Disambiguation Models for Wide–Coverage HPSG parsing. In *ACL–2005*, 2005.
11. I. Sag, T. Wasow, and E. Bender. *Syntactic Theory -A Formal Introduction. Second Edition.* CSLI PUBLICATIONS, 2003.
12. L. Steels and J.D Beule. A (very) Brief Introduction to Fluid Construction Grammar. In *Proceedings of the 3rd International Workshop on Scalable Natural Language (ScaNaLu06)*, to appear.