

Incremental Sentence Production with a Parallel Marker-Passing Algorithm

Hiroaki Kitano
Center for Machine Translation
Carnegie Mellon University
Pittsburgh, PA 15213, U.S.A.
hiroaki@cs.cmu.edu

ABSTRACT

This paper describes a method of incremental natural language generation using a parallel marker-passing algorithm for modeling simultaneous interpretation. Semantic and syntactic knowledge are represented in a memory network in which several types of markers are passed around in order to make inference, and explore implicit parallelism of sentence production. The model is consistent with several psycholinguistic studies. The model is actually implemented as a part of the Φ DMDIALOG real-time speech-to-speech dialog translation system developed at the Center for Machine Translation at Carnegie Mellon University, and publicly demonstrated since March 1989.

1 Introduction

Incremental sentence production has been gaining more attention in recent years. It is particularly important in application areas such as speech-to-speech translation where real-time transaction is essential. The Φ DMDIALOG project is a research project to develop a speech-to-speech dialog translation system with simultaneous interpretation capability. At the outset of the project, we have investigated actual simultaneous interpretation sessions and telephone dialogs. As a result, we found that one utterance in a real dialog can be quite long (15 seconds for one sentence is not rare in Japanese). This implies that if we adopt a sequential architecture, in which generation starts only after the entire parsing is completed, this inevitably creates unendurable delay in translation. Suppose one speaker made an utterance of 15 seconds, and the other responded with an utterance of 15 seconds in length, the first speaker must wait at least 30 seconds to start hearing the translation of the utterance of his/her dialog partner. It is inconceivable that such a system could be practically deployed. Introduction of a simultaneous interpretation scheme by coupling an incremental generation and an incremental parsing technologies is the only way to minimize this problem¹.

Incremental sentence production is interesting from the standpoint of psycholinguistics as well. There are many psycholinguistic studies which support incremental sentence production as a psycholinguistically plausible approach. We will discuss the psycholinguistic relevancy of our model later.

In this paper, we describe a model of incremental sentence production which is actually implemented as a part of the speech-to-speech dialog translation system

¹Although there is a problem of how to resolve ambiguities in parsing, discussion on such topic is beyond the scope of this paper. For those who are interested in this topic, refer to [Kitano et. al., 1989a][Kitano et. al., 1989b].

Φ DMDIALOG [Kitano, 1989a], developed at the Center for Machine Translation at Carnegie Mellon University.

2 Basic Organization of the Model

We use a hybrid parallel paradigm [Kitano, 1989b], which is an integration of a parallel marker-passing scheme and a connectionist network, as a basic algorithm. Five types of markers (two types for parsing, two other types for generation, and an another type for contextual priming) are passed around the memory network which represents knowledge from morphophonetic-level to discourse-level. A connectionist network performs sub-symbolic computations with a massive parallelism. Use of the hybrid parallel scheme on the memory network has its merit in exploring implicit parallelism in the process of natural language generation and parsing.

2.1 The Memory Network

The memory network incorporates knowledge from morphophonetics to plan hierarchies of each participant of a dialog. Each node is a *type* and represents either a concept (Concept Class node; CC) or a sequence of concepts (Concept Sequence Class node; CSC). Strictly speaking, both CC and CSC are a *collection* or *family* since they are, for the most part, sets of classes. CCs represent such knowledge as concepts (i.e. *Conference, *Event, *Mtransaction), and plans (i.e. *Declare-Want-Attend). CSCs represent sequences of concepts and their relations such as concept sequences² (i.e. <*Conference *Goal-Role *Attend *Want>) or plan sequences (i.e. <*Declare-Want-Attend *Listen-Instruction>)³ of the two participants of the dialog. CSCs have an internal structure composed of a concept sequence, constraint equations, presuppositions, and effects. This internal structure provides our scheme with the capability to handle unification-based processing as well as case-based processing, so that typical criticisms against DMAP-type NLP [Riesbeck and Martin, 1985], such as weak linguistic coverage and incapability of handling linguistically complex sentences, do not apply to our model⁴. Each type of node creates instances during parsing which are called concept instances (CI) and concept sequence instances (CSI), respectively. CIs correspond to discourse entities. They are connected through labelled links such as IS-A or PART-OF, and weighted links which form a connectionist network. CSIs record specific cases of utterances indexed into the memory network whereas

²Concept sequences are the representation of an integrated syntax/semantics level of knowledge in our model.

³This should not be confused with 'discourse segments' [Grosz and Sidner, 1985]. In our model, information represented in discourse segments is distributively incorporated in the memory network.

⁴Indeed, our model is substantially different from DMAP-type marker-passing or any other naive marker-passing models, because linguistic features are carried up by markers to conduct substantial linguistic analysis as well as case-based processing.

CSCs represent generalized cases and syntactic rules. Use of cases for generation is one of the unique features of our model while most generators solely depend upon syntactic rules.

2.2 The Markers

A guided marker-passing scheme is employed for inference in the memory network. Basically, our model uses four types of markers. These markers are (1) activation markers, (2) prediction markers, (3) generation markers, and (4) verbalization markers.

Activation Markers (A-Markers) are created based on the input of the source language. These are passed up through IS-A links and carry instance, features and cost. This type of marker is used for parsing.

Prediction Markers (P-Markers) are passed along the conceptual and phonemic sequences to make predictions about which nodes are to be activated next. Each P-Marker carries constraints, cost, and the information structure of the utterance which is built incrementally during parsing.

Generation Markers (G-Markers) show activation of nodes in the target language, and each contains a surface string, features, cost and an instance which the surface string represents. G-Markers are passed up through IS-A links.

Verbalization Markers (V-Markers) anticipate and keep track of verbalization of surface strings. Final surface realizations, cost and constraints are carried by V-Markers.

Besides these markers, we assume **Contextual Markers (C-Markers)** [Tomabechi, 1987] which are used when a connectionist network is computationally too expensive. The C-Markers are passed through weighted links to indicate contextually relevant nodes.

2.3 A Baseline Algorithm

Generally, natural language generation involves several stages: content delimitation, text structuring, lexical selection, syntactic selection, coreference treatment, constituent ordering, and realization. In our model, the content is determined at the parsing stage, and most other processes are unified into one stage, because, in our model, lexical item, phrase, and sentence are treated in the same mechanism. The common thrust in our model is the hypothesis-activation-selection cycle in which multiple hypotheses are activated and where one of them is finally selected. Thus, the translation process of our model is composed of processes of (1) concept activation, (2) lexical and phrasal hypotheses activation, (3) propositional content activation, (4) syntactic and lexical selection, and (5) realization.

1. Concept Activation: A part of the parsing process as well as an initial process of generation. Individual concepts represented by CCs are activated as a result of parsing speech inputs. A-Markers are created and passed up by activating the concept.

2. Lexical and Phrasal Hypotheses Activation: Hypotheses for lexicons and phrases which represent the activated concept are searched for, and G-Markers are created and passed up as a result of this process. Usually, multiple candidates are activated at a time.

3. Propositional Content Activation: A part of the parsing process by which propositional content of the utterance is determined.

4. Syntactic and Lexical Selection: Selection of one hypothesis from multiple candidates of lexical entries or phrases. First, the syntactic and semantic constraints are checked to ensure the correctness of the hypotheses, and the final selection is made using a cost/activation-based selection.

5. Realization: The surface string (which can be either a sequence of words or a sequence of phonological signs) is formed from the selected hypothesis and sent to the speech synthesis device.

The movement of V-Markers is important in understanding our algorithm. First, a V-Marker is located on the first element of the CSC. When a G-Marker hits the element with the V-Marker, the V-Marker is moved to the next element of the CSC (figure 1a), and unification is performed to ensure syntactic soundness of the sentence. In figure 1b, \bar{a}_1 is a *closed class lexical item*⁵. When a G-Marker hits the first element, a V-Marker on the first element is moved to the third element by passing through the second element which is a closed class item. In this case, the element for the closed class item need not have a G-Marker. The lexical realization for the element is retrieved when the V-Marker passes through the element. In the case where the G-Marker hits an element without a V-Marker, the G-Marker is stored in the element. When another G-Marker hits the element with a V-Marker, the V-Marker is moved to the next element. Since the next element already has a G-Marker, the V-Marker is further moved to the subsequent element of the CSC (figure 1c). Although, in most cases, a bottom up process by G-Markers handles generation process, there are cases where a bottom up process alone can not identify syntactic structure and lexical items to express a given meaning. In such cases, a top-down process is invoked which identifies the best syntactic structure and lexical items by searching downward from each element of the activated CSC. Each retrieval procedure is similar to the search of a closed class lexical item.

There are cases in which an element of the CSC is linked to other CSCs, and forms hierarchies of CSCs. Suppose each CSC represents a phrase structure rule, then the dynamically organized CSC hierarchy provides productive power so that various types of structures of complex sentences can be generated. In the hierarchy of CSCs, G-Markers are passed up when a CSC is accepted, and carry feature structures which represent meaning fragments expressed by the CSC. V-Markers are passed down to lower CSCs when an element is predicted, and impose constraints on each elements of the lower CSCs. The hierarchical organization of CSCs allows all types of tree expansions: upward, downward and insertion.

Figure 2 shows an example of how an analysis tree can be constructed in our model. In this example, we assume Lexical-Functional Grammar (LFG) as a grammar formalism, and the order which conceptual fragments are given is based on an order that conceptual fragments can be identified when parsing a corresponding Japanese sentence incrementally. Notice that all three types of extensions are involved even in such a simple sentence.

⁵*Closed class lexical items refer to function words such as in, of, at in English and wo, ga, ni in Japanese. These words are non-referential and their number do not grow, whereas open class lexical items are mostly referential and their number grows as vocabulary expands.*

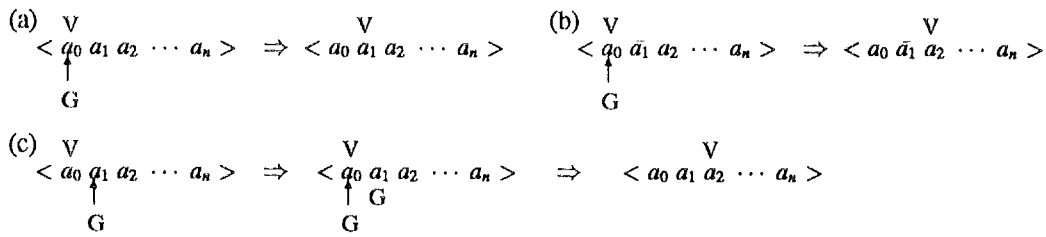


Figure 1: Movement of V-Marker in the CSC

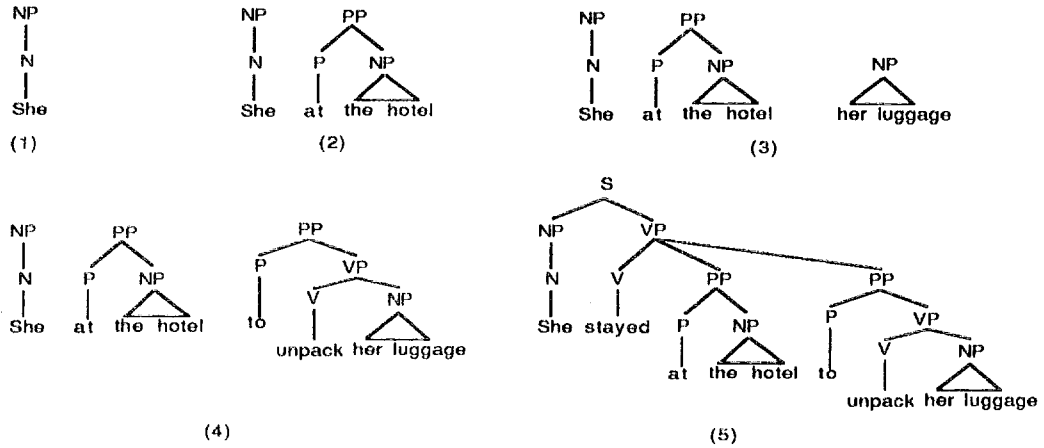


Figure 2: An Incremental Tree Construction

3 Activation of Lexical and Phrasal Hypotheses and Propositional Contents

When a concept is recognized by the parsing process, an hypotheses for translation will be activated. The concept can be an individual concept, a phrase or a sentence. In our model, they are all represented as CC nodes, and each instance of the concept is represented as a CI node. The basic process is for each of the activated CCs, LEX nodes⁶ in the target language to be activated. There are four possible mappings between source language nodes and target language nodes which are activated; word-to-word, phrase-to-word, word-to-phrase, and phrase-to-phrase. In our model, hypotheses for sentences and phrases are represented as CSCs. From the viewpoint of generation, either LEX nodes representing words or CSC nodes representing phrases or entire sentences are activated.

LEX node activation: There are cases when a word or a phrase can be translated into a word in the target language. In figure 3a and c, the word LEX_{SL} or the phrase CSC_{SL} activates CC_1 . LEX_{1TL} is activated as a hypothesis of translation for LEX_{SL} or CSC_{SL} interpreted as CC_1 . A G-Marker is created at LEX_{TL} containing a surface realization, cost, features, and an instance which the LEX_{1TL} represents (CI). The G-Marker is passed up through an IS-A link. When a CC_1 does not have LEX_{1TL} , a CC_2 is activated and a LEX_{2TL} will be activated. Thus, the most specific word in the target language will be activated as a hypothesis.

CSC node activation: When a CC can be represented by a phrase or sentence, a CSC node is activated and a

⁶LEX nodes are a kind of CSC which represent a lexical entry and phonological realization of the word.

G-Marker which contains that phrase or sentence will be created. In figure 3b and d, LEX_{SL} and CSC_{SL} activates CC_1 which has CSC_{1TL} . In this case, CSC_{1TL} will be activated as a hypothesis to translate LEX_{SL} or CSC_{SL} interpreted as CC_1 . In particular, activation of CSC_{TL} by CSC_{SL} is interesting because it covers cases where two expressions can be translated only at phrasal or sentential correspondence, not at the lexical level. Such cases are often found in greetings or canned phrases. It should be noted that CSCs represent either syntactic rules or cases of utterance. Assuming cases are acquired from legitimate utterances of native speakers, use of cases for a generation process should be preferred over purely syntactic formulation of sentences because use of cases avoids generation of sentences which are syntactically sound but never uttered by native speakers.

4 Syntactic and Lexical Selections

Syntactic and lexical selections are conducted involving three processes: feature aggregation, constraint satisfaction, and competitive activation. Feature aggregation and constraint satisfaction correspond to a symbolic approach to syntactic and lexical selection which guarantee grammaticality and local semantic accuracy of the generated sentences, and the competitive activation process is added in order to select the best decision among multiple candidates.

4.1 Feature Aggregation

Feature aggregation is an operation which combines features in the process of passing up G-Markers so that minimal features are carried up. Due to the hierarchical organization of the memory network, features which

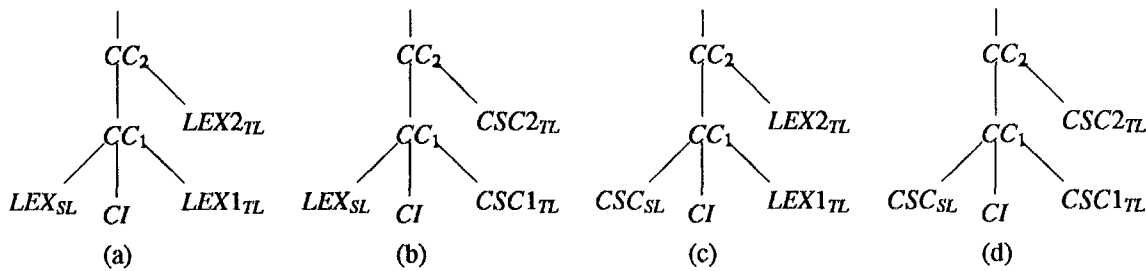


Figure 3: Activation of Syntactic and Lexical Hypotheses

need to be carried by G-Markers are different depending upon which level of abstraction is used for generation⁷. Given the fact that unification is a computationally expensive operation, aggregation is an efficient mechanism for propagating features because it ensures only minimal features are aggregated when features are unified, and aggregation itself is a cheap operation since it simply adds new features to existing features in the G-Marker. One other advantage of this mechanism is that the case-based process and the constraint-based process are treated in one mechanism because features required for each level of processing are incrementally added in G-Markers.

4.2 Constraint Satisfaction

Constraint is a central notion in modern syntax theories. Each CSC has *constraint equations* which define the constraints imposed for that CSC depending on their level of abstraction⁸. Feature structures and constraint equations interact at two stages. At the prediction stage, if a V-Marker placed on the first element of the CSC already contains a feature structure that is non-nil, the feature structure determines, according to the constraint equations, possible feature structures of G-Markers which subsequent elements of the CSC can accept. At a G-V-collision stage, a feature structure in the G-Marker is tested to see if it can meet what was anticipated. If the feature structure passes this test, information in the G-Marker and the V-Marker are combined and more precise predictions are made as to what will be acceptable in subsequent elements. Thus, the grammaticality of the generated sentences is guaranteed. Semantic restrictions are considered in this stage.

4.3 Competitive Activation

The competitive activation process introduced either by a C-Marker-passing or by the connectionist network determines the final syntactic and lexical realization of the sentence. Here, we have adopted a cost-based scheme as we have employed in parsing [Kitano et. al., 1989a]. In the cost-based scheme, the hypothesis with the least cost will be selected. This idea reflects our view that parsing and generation are dynamic processes in which the state of the system tends to a global minima, and that a cost represents dispersion of energy so that higher cost hypotheses are less likely to be taken as the state of the system. In the actual implementation, we compute a cost

of each hypothesis which is determined by a C-Marker-passing scheme or a connectionist network.

The C-Marker passing scheme puts C-Markers at contextually relevant nodes when a conceptual root node is activated. A G-Marker which goes through a node without a C-Marker will be added with larger cost than others. When there are multiple hypothesis for the specific CC node; i.e. when multiple CSCs are linked with the CC, we will add up the cost of each G-Marker used for each linearization combined with pragmatic constraints which may be assigned to each CSC, and the preference for each CSC, and the hypothesis with least cost will be selected as the translated result.

The Connectionist Network will be adopted with some computational costs. When a connectionist network is fully deployed, every node in the network is connected with weighted links. A competitive excitation and inhibition process is performed to select one hypothesis. Final interpretation and translation in the target language are selected through a winner-take-all mechanism.

5 Commitment and Ambiguities

One of the most significant issues is how to resolve ambiguities of the parsing process as early as possible, so that the final translation hypothesis can be determined as early as possible. Since many sentences are ambiguous until, at least, the entire clause is analyzed, disambiguation necessarily imposes constraints upon scheduling of the generation process. However, it should be noted that the human interpreter does not start translating unless she/he is sure about what the sentence means. This allows our model to take a wait-and-see strategy when multiple hypotheses are present during processing of input utterances. However, when some ambiguities still remain, the generator needs to commit to one of the hypotheses, which may turn out to be false. This would be even complicated when a source language and a target language have substantially different linguistic structures. For example, in English, negation comes before a verb, whereas Japanese negation comes after a verb, and the verb comes at the very end of a sentence. In such case, translation cannot be started until the verb, which comes the end of the sentence, was processed, and existence of negation after the verb is checked. Decision has to be made, for this case, to wait translation until these ambiguities are resolved by encountering a clause which follows the initial clause. Fortunately, most Japanese utterance consist of multiple clauses which makes simultaneous interpretation possible. In order to cope with these ambiguities, a simultaneous interpretation system should have capabilities such as (1) anticipating the possibility of negation at the end, (2) incorporating some heuristics which recover

⁷When knowledge of cases, similar to the *phrasal lexicon*, is used for generation, features are not necessary because this knowledge is already indexed to specific discourse entities.

⁸However, CSCs representing specific cases do not have constraint equations since they are already instantiated and the CSCs are indexed in the memory network.

false translation to correct one, and (3) making decisions on when to start or wait translations. Theories of commitment in ambiguity resolution and generation are not established, yet, thus they are a subject of further investigations. One possible solution which we are investigating is to use probabilistic speed control of marker propagation as seen in [Wu, 1989] so that the best hypothesis is presented first. This would allow the generator to commit upon present hypothesis within its local decisions.

6 Psychological Plausibility

Psychological studies of sentence production [Garrett, 1975] [Garrett, 1980] [Levelt and Maassen, 1981] [Bock, 1982] [Bock, 1987] and [Kempen and Huijbers, 1983] were taken into account in designing the model. In [Kempen and Huijbers, 1983], two independent retrieval processes are assumed, one accounting for abstract pre-phonological items (L1-items) and the other for phonological items (L2-items). The lexicalization in their model follows: (1) a simultaneous multiple L1-item retrieval, (2) a monitoring process which watches the output of L1-lexicalization to check that it is keeping within constraints upon utterance format, (3) retrieval of L2-items after waiting until the L1-item has been checked by the monitor, and all other L1-items become available. In our model, a CCs activation stage corresponds to multiple L1-item retrieval, constraint checks by V-Markers correspond to the monitoring, and the realization stage which concatenates the surface string in a V-Marker corresponds to the L2-item retrieval stage. The difference between our model and their model is that, in our model, L2-items are already incorporated in G-Markers whereas they assume L2-items are accessed only after the monitoring. Phenomenologically, this does not make a significant difference because L2-items (phonological realization) in our model are not explicitly selected until constraints are met; at which point the monitoring is completed. However, this difference may be more explicit in the production of sentences because of the difference in the scheduling of the L2-item retrieval and the monitoring. This is due to the fact that our model retains interaction between two levels as investigated by [Bock, 1987]. Our model also explains contradictory observations by [Bock, 1982] and [Levelt and Maassen, 1981] because activation of CC nodes (L1-items) and LEX nodes (L2-items) are separated with some interactions. Also, our model is consistent with a two-stage model [Garrett, 1975] [Garrett, 1980]. The *functional* and *positional* levels of processing in his model correspond to the parallel activation of CCs and CSCs, the V-Marker movement which is left to right, and the surface string concatenation during that movement.

Studies of the planning unit in sentence production [Ford and Holmes, 1978] give additional support to the psychological plausibility of our model. They report that *deep clause* instead of *surface clause* is the unit of sentence planning. This is consistent to our model which employs CSCs, which account for deep propositional units and the realization of deep clauses as the basic units of sentence planning. They also report that people are planning the next clause while speaking the current clause. This is exactly what our model is performing, and is consistent with our observations from transcripts of simultaneous interpretation.

7 Relevant Studies

Since most machine translation systems assume sequential parsing and generation, a simple extension of existing systems to combine speech recognition and synthesis would not suffice for interpreting telephony. The main problem is in previously existing systems' inability to attain simultaneous interpretation (whereas partial translation is performed while parsing is in progress), because in other systems a parser and a generator are independent modules, and the generation process is only invoked when the entire parse is completed and full semantic representation is given to the generator. Our model serves as an example of approaches counter to the modular approach, and attains simultaneous interpretation capability by employing incremental parsing and a generation model. Pioneer studies of parallel incremental sentence production are seen in [Kempen and Hoekamp, 1987] [Kempen, 1987]. They use a segment grammar which is composed of Node-Arc-Node building blocks to attain incremental formation of trees. Their studies parallel our model in many aspects. The segment grammar is a kind of semantic grammar since the arc label of each segment makes each segment a syntax/semantic object. Feature aggregation and constraint satisfaction by G-Markers and V-Markers in our model corresponds to a distributed unification [De Smedt, 1989] in the segment grammar. [De Smedt, 1990] reports extensively on their approach to incremental sentence generation which parallel to our model in many aspects.

8 Current Implementation

The model of generation described in this paper has been implemented as a part of Φ DMDIALOG, a speech-to-speech dialog translation system developed at the Center for Machine Translation at Carnegie Mellon University. Φ DMDIALOG is implemented on an IBM RT-PC workstation using CMU CommonLisp run on Mach OS. Speech input and voice synthesis are done by connected hardware systems, currently, we are using Matsushita Institute's Japanese speech recognition hardware and DECTalk.

Figure 4 is an example of how sentences with multiple clauses are translated simultaneously in Φ DMDIALOG. Although an input is shown as a word sequence, real run takes speech inputs and a phoneme sequence is used to interface between the speech recognition device and the software. Current implementation translates between Japanese and English and operates on the conference registration domain based on the corpus provided by the ATR Interpreting Telephony Research Laboratories. For more details of the generation scheme described in this paper, refer to [Kitano, 1990].

Currently, we are designing a version of our model to be implemented on massively parallel machines: IXM [Higuchi et. al., 1989] and SNAP [Moldovan et. al., 1989].

9 Conclusion

We described a parallel incremental model of natural language generation designed for the speech-to-speech dialog translation system Φ DMDIALOG. We demonstrated that a parallel marker-passing scheme is one desirable way of exploring inherent parallelism of sentence production. All types of tree expansion are attained, and ability to in-

Input Utterance	Translation
I want to attend the conference because	watashi ha (I Role-Agent; This is ellipsed in the actual translation) kaigi ni sanku shitai (want to attend the conference)
I am interested in interpreting telephony	toiunoha (because) watashi ha (I Role-Agent; This is ellipsed in the actual translation) tuuyaku denwa ni kyoomi ga arukara desu (interested in interpreting telephony)

Figure 4: An Example of Simultaneous Interpretation

crementally generate complex sentences has been shown. It should be noted that, in our model, activations and selections of syntactic structure and lexical items are treated in a uniform mechanism. Psychological plausibility is another notable feature of our model since most research in natural language generation has not taken into account psychological studies. We believe our parallel incremental generation model is a promising approach toward the development of interpreting telephony where simultaneous interpretation is required.

References

- [Bock, 1987] "Exploring Levels of Processing in Sentence Production," In Kempen, G. (Ed.) *Natural Language Generation*, Nijhoff, 1987
- [Bock, 1982] "Toward a Cognitive Psychology of Syntax: Information Processing Contributions to Sentence Formulation," *Psycho. Rev.*, 89, pp1-47, 1982.
- [De Smedt, 1989] De Smedt, K., "Distributed Unification in Parallel Incremental Syntactic Tree Formation," In *Proceedings of the Second European Workshop on Natural Language Generation*, 1989.
- [De Smedt, 1990] De Smedt, K., *Incremental Sentence Generation*, NCI Technical Report 90-01, Nijmegen Institute for Cognition Research and Information Technology, 1990.
- [Ford and Holmes, 1978] Ford, M. and Holmes, V., "Planning Units and Syntax in Sentence Production," *Cognition*, 6, pp35-53, 1978.
- [Garrett, 1980] Garrett, M.F., "Levels of Processing in Sentence Production," In Butterworth, B. (Ed.) *Language Production (Vol. 1 Speech and Talk)*, Academic Press, 1980.
- [Garrett, 1975] Garrett, M.F., "The Analysis of Sentence Production," In Bower, G. (Ed.) *The Psychology of Learning and Motivation*, Vol. 9, Academic Press, 1975.
- [Grosz and Sidner, 1985] Grosz, B. and Sidner, C., "The Structure of Discourse Structure," *CSLI Report No. CSLI-85-39*, 1985.
- [Higuchi et. al., 1989] Higuchi, T., Furuya, T., Kusumoto, H., Handa, K. and Kokubu, A., "The Prototype of a Semantic Network Machine IXM," In *Proceedings of the International Conference on Parallel Processing*, 1989.
- [Kempen, 1987] Kempen, G., "A Framework for Incremental Syntactic Tree Formation," In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-87)*, 1987.
- [Kempen and Hoekamp, 1987] Kempen, G. and Hoekamp, E., "An Incremental Procedural Grammar for Sentence Formulation," *Cognitive Science*, 11, 201-258, 1987.
- [Kempen and Huijbers, 1983] Kempen, G. and Huijbers, P., "The Lexicalization Process in Sentence Production and Naming: Indirect Election of Words," *Cognition*, 14, pp185-209, 1983.
- [Kitano, 1990] Kitano, H., "Parallel Incremental Sentence Production for a Model of Simultaneous Interpretation," In Dale, R. et. al. (Eds.) *Current Research in Natural Language Generation*, Academic Press, 1990.
- [Kitano, 1989a] Kitano, H., *A Massively Parallel Model of Simultaneous Interpretation: The Φ DMIALOG System*, Technical Report CMU-CMT-89-116, Carnegie Mellon University, Pittsburgh, 1989.
- [Kitano, 1989b] Kitano, H., "Hybrid Parallelism: A Case of Speech-to-Speech Dialog Translation," In *Proceedings of the IJCAI-89 Workshop on Parallel Algorithms for Machine Intelligence*, 1989.
- [Kitano et. al., 1989a] Kitano, H., Tomabechi, H. and Levin, L., "Ambiguity Resolution in DMTRANS PLUS," In *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, 1989.
- [Kitano et. al., 1989b] Kitano, H., Mitamura, T. and Tomita, M., "Massively Parallel Parsing in Φ DMIALOG: Integrated Architecture for Parsing Speech Inputs," In *Proceedings of the International Workshop on Parsing Technologies*, 1989.
- [Levelt and Maassen, 1981] Levelt, W.J.M. and Maassen, B., "Lexical Search and Order of Mention in Sentence Production," In Klein, W. and Levelt, W.J.M. (Eds.), *Crossing the Boundaries in Linguistics: Studies Presented to Manfred Bierwisch*, Dordrecht, Reidel, 1981.
- [Moldovan et. al., 1989] Moldovan, D., Lee, W. and Lin, C., *SNAP: A Marker-Propagation Architecture for Knowledge Processing*, Technical Report CENG 89-10, University of Southern California, 1989.
- [Riesbeck and Martin, 1985] Riesbeck, C. and Martin, C., *Direct Memory Access Parsing*, *Yale University Report 354*, 1985.
- [Tomabechi, 1987] Tomabechi, H., "Direct Memory Access Translation," *Proceedings of IJCAI87*, 1987.
- [Wu, 1989] Wu, D., "A Probabilistic Approach to Marker Propagation" In *Proceedings of IJCAI-89*, 1989.