(C5)

SPEECHTRANS:
An Experimental Real-Time Speech-to-Speech Translation System[1]

Hideto Tomabechi, Hiroaki Saito, and Masaru Tomita
Center for Machine Translation
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This paper reports the current progress in the SPEECHTRANS project at the Center for Machine Translation which is a speech-to-speech translation project for real-time processing of speaker-independent noisy continuous speech input. SPEECHTRANS uses a custom speech recognition hardware and a phoneme-based generalized LR parser that uses a unification-based grammar formalism and a natural language generator that is connected to a voice synthesis module. SPEECHTRANS was originally presented at the MT Conference held at CMU in June 1988 as a public demonstration of a real-time speaker-independent speech-to-speech translation system.

1 Introduction

Efforts in the areas of speech recognition, natural language understanding, generation and voice synthesis have come to the point that we can now integrate them into a real-time speech processing system. At the Center for Machine Translation (CMT) at Carnegie Mellon University, we have combined a custom-made speech recognition hardware with our machine translation systems developing a phoneme-based parser and connecting a generator output to an existing voice synthesis module. The SPEECHTRANS system, which we built by connecting two AI workstations with recognition and synthesis hardware as a real-time speech translation system, was publicly demonstrated at the MT Conference sponsored by CMT in June 1988. The system is speaker-independent and performs speech-to-speech translation at real-time. We will be reporting the progress in this project at CMT in this paper, including our findings for need for further research.

2 The Modularity of SPEECHTRANS

The system is modular and consists of four parts:

- Speech recognition hardware;
- Phoneme-based syntax/semantics/pragmatics recognizer;
- Natural language generator;
- Speech synthesis module.

The first and the last modules were built outside of CMT, namely, the recognition hardware was custom built by Matsushita Research Institute (Morii, *et al*[1985]) and the speech synthesis module (DECTALK) by Digital Equipment Corporation. The phoneme-based parser is a phoneme-based generalized LR parser (Saito&Tomita[1988b]) that is designed to handle noise in the input and uses a unification-based (LFG[2]) integrated (syntax/semantics) parsing scheme. The natural language generator uses a pseudo-unification grammar adopting the precompilation method used in the generalized LR parser.

Due to the modularity of the system, efforts are underway to replace some of the modules with different systems. One such effort is to use SPHINX (Lee[1988]) system for English speech input.

3 Speech Recognition Hardware

The input to the system is a sequence of phonemes. The custom built speech recognition device takes a continuous speech utterance, for example 'megaitai' ('I have a pain in my eye.'), from a microphone and produces a *noisy* phoneme sequence such as 'ebaitaai'[3].

The speech recognition device does not have any syntactic nor semantic knowledge and produces a phoneme sequence (noisy), not a phoneme lattice; there are no other phoneme candidates available to alternate. We must make the best guess based solely on the phoneme sequence generated by the speech device. Errors caused by the speech device can be classified into three groups:

- Substituted Phonemes – Phonemes recognized incorrectly. The second phoneme /b/ in 'ebaitaai' is a substituted phoneme, for example.

[1]Funding for this project is provided by several private institutions and governmental agencies in the United States and Japan. Parts of this paper have appeared in Saito&Tomita[1988a/b] and Tomabechi&Tomita[1988a].

[2]Lexical Functional Grammar (Kaplan&Bresnan[1982]).

[3]We distinguish *noisy* from *ill-formed*. The former is due to recognition device errors, while the latter is due to human users.

- Deleted Phonemes – Phonemes not recognized by the device which are actually spoken. For example a phoneme /m/ is missed at the beginning of 'ebaitaai.'
- Inserted Phonemes – Phonemes recognized by the device which are not actually spoken. The penultimate phoneme, /a/, in 'ebaitaai' is an inserted phoneme, for example.

To cope with these problems, we need:

1. A very efficient parsing algorithm, as our task requires much more search than conventional typed sentence parsing.
2. A good scoring scheme, to select the most likely hypothesis out of multiple candidates.
3. Syntactic, semantic, and pragmatic constraints to narrow down candidate groupings of streams of phonemes.

In the next section we describe the parsing algorithm and the scoring scheme. Also, a discussion of use of pragmatic knowledge at real-time using a massively-parallel network of contextual memory will be discussed in the later section of this paper.

4 Phoneme-based GLR Parser

In this section, we describe the Phoneme-based Generalized LR Parser (ΦGLR) that is used in our system. The parser is based on the Universal Parser Architecture (Tomita[1985]) with an added scheme to work on a stream of phonemes instead of text. We have two versions of the parser running in our SpeechTrans system: One that utilizes modularized syntax (LFG) and semantic (case-frame) knowledge, merging them at run-time, and another version which uses a hand-coded grammar with syntax and semantics precompiled into one pseudo-unification grammar. The former is our standard system; however, the latter is often used for demonstration purposes because of added speed at run-time. In this section, we will be concentrating on the scheme that we adopted to use the generalized LR parsing algorithm to handle noisy speech input. The grammar we are using is an Augmented Context-Free Grammar whose terminal symbols are phonemes rather than words. That is, the grammar includes rules like:

Noun ⟶ /w/ /a/ /t/ /a/ /s/ /i/

instead of

Noun ⟶ "watasi".

The morphological and syntactic grammar (Tomita, *et al*[1987]) has been developed primarily for CMU's knowledge-based machine translation system (Tomita&Carbonell-[1987]), and it consists of more than 2000 rules including lexical rules like the one above with the addition of phonemic knowledge.

4.1 LFG-based syntax and semantic mappings

The syntactic formalism that is used in our SpeechTrans project is motivated by LFG[4]. LFG is one of the unification-based grammar formalisms in which a method of combining partial information is implemented through unification of feature-structures (attribute value matrices). LFG models a posited level of syntactic representation called f-structure (functional structure) which contains information about the grammatical relations of an expression as a result of unification of partial informational structures supplied by the grammar and the lexicon. In LFG, unlike HPSG (Head-driven Phrase Structure Grammar – Pollard&Sag[1987]), semantic contents of signs are not part of the informational structures built by the grammar and the input. In our project, LFG is used strictly for syntactic processing (i.e., LFG is used as a syntactic formalism) and the semantic content of signs are handled through syntax/semantics mapping rules that map specific semantic relations (according to a given domain knowledge) to LFG-like grammatical relations. The syntactic knowledge and the semantic knowledge are maintained separately and are precompiled and automatically merged[5] to generate a run-time augmented[6] context-free grammar (ACFG) which is further compiled automatically into an augmented LR parsing table[7] which will be used by the phoneme-based generalized LR parser. The original SpeechTrans system used pseudo-unification as a base of grammatical operations due to the run-time speed considerations; however, we have also supported full-unification based grammar by using a fast non-destructive graph unification algorithm (Wroblewski[1987]).

4.2 Handling Substituted, Inserted, and Deleted Phonemes

Tomita[1985] introduced the *Generalized LR Parsing Algorithm* for Augmented Context-Free Grammars, which can handle nondeterminism and ambiguity using *graph-structured stacks*. We modified the algorithm to receive phonemes as input and to cope with altered, extra and missing phonemes while parsing an input from left to right. Specifically modifications were made to handle the phenomena of noisy input as below:

---

[4]Although, it is not strictly LFG, especially because we use pseudo-unification instead of full-unification.

[5]To avoid misunderstanding, we should clarify that the semantic knowledge itself does not get merged into the ACFG, instead, the constraints on unification (or pseudo-equations in case of pseudo-unification) which trigger semantic processing are merged into the augmentations.

[6]Augmentation is necessary in order to capture information-combining unification operation that unification-based grammar formalisms require. Also, compilation from unification-based grammar formalism into ACFG is done automatically by default; however, for the original version of SpeechTrans, we hand-compiled the grammar and the semantics because of the initial efficiency considerations and the close control of the parsing processes.

[7]This scheme is explained in detail in Tomita[1985].

- **Substituted phonemes:** substituted and thus may be incorrect. The parser has to consider all these possibilities. We can create a phoneme lattice dynamically by placing alternate phoneme candidates in the same location as the original phoneme. Each possibility is then explored by each branch of the parser. Not all phonemes can be altered to any other phoneme. For example, while /o/ can be mis-recognized as /u/, /i/ can never be mis-recognized as /o/. This kind of information can be obtained from a *confusion matrix*, which we shall discuss in the next subsection. With the confusion matrix, the parser does not have to exhaustively create alternate phoneme candidates .
- **Inserted phonemes:** Each phoneme in a phoneme sequence may be an extra, and the parser has to consider these possibilities. We have one branch of the parser consider an extra phoneme by simply ignoring the phoneme. The parser assumes at most *two* inserted phonemes can exist between two real phonemes, and we have found the assumption quite reasonable and safe.
- **Deleted phonemes:** Deleted phonemes can be handled by inserting possible deleted phonemes between two real phonemes. The parser assumes that at most one phoneme can be missing between two real phonemes.

4.3 Scoring and the Confusion Matrix

Tomita[1986] introduces a scheme for word lattice parsing based on the generalized LR parsing architecture, which is the basis of our phoneme-based parsing scheme. In this modified scheme we use the mechanism of scoring each parse based upon a confusion matrix of phonemes. There are two main reasons why we want to score each parse: first, to prune the search space by discarding branches of the parse whose score is hopelessly low; second, to select the best sentence out of multiple candidates by comparing their scores.

Branches of the parse which are accompanied with fewer substituted/inserted/deleted phonemes should be given her scores. Whenever a branch of the parse handles a substituted/inserted/deleted phoneme, a specific penalty is given to the branch. Scoring accuracy can improve with the confusion matrix.

Two methods have been adopted to prune partial parses by a score:

- Discarding the low-score shift-waiting branches when a phoneme is applied.
- Discarding the low-score branches in a local ambiguity packing.

The former method is very effective when strictly applied.

The confusion matrix only shows us the phoneme-to-phoneme transition, therefore a broader unit transition should also be considered, such as a tendency for the /w/ phoneme in 'owa' or 'owo' to be missed or for the very first /h/ sound of an input to be missed, and the frequent transformation to 'h@[8]' of the 'su' sound in 'desuka.'

5 Current Issues in SPEECHTRANS

5.1 Confusion Matrix adds complexity

When compared to the understanding of text input, the added difficulty in the understanding of continuous speech input can be seen in: **phonemic segmentation, added lexical ambiguity**, and **extra-grammaticality and incomplete words** due to the noise and the connectedness of the input[9]. We use the confusion matrix and the schemes introduced in the previous section to tackle the problems of missing, added, altered phonemes as well as the occurrance of allophomic variations. On one hand, this is necessary in order to handle noisy[10] input. On the other hand, this makes the number of acceptable hypotheses for a given input extremely large. Our scheme is to parse the noisy input rather generously with the schemes introduced in the previous section in order to cover the weakness of incomplete output of the speech recognition hardware as well as the noise in the environment. Our preliminary version of the SPEECHTRANS system countered this problem through limiting the size of the grammar and the vocabulary so that only a small number of (syntactic/semantic) ambiguities will result after the parse. This is sufficient as a prototype speech-to-speech translation system for public demonstration; however, in order to make the grammar and the vocabulary sufficiently large to cover the practical domain, the problem of the rapid growth of acceptable ambiguities often becomes intolerable.

Local semantic restriction checks are not sufficient for disambiguating continuous speech input, since an interpretation can be totally legitimate semantically, but can mean something drastically different from what has been input into the speech recognition system (as well as being contextually inappropriate). The speech understanding system needs extra-sentential knowledge to choose an appropriate hypothesis for grouping phonetic segments and for selecting the appropriate word-sense of lexical entries. In other words, the need for contextual knowledge in speech understanding systems is even more urgent than in text input understanding systems; in a speech understanding system, the input can be interpreted in a way that is not possible in text input systems, and the input can still be acceptable to the local semantic restriction checks that integrated parsers perform within a sentence (such as slot-filler restriction checks of case-frame parsers).

[8] @ represents that it is a vowel which may be either /i/ or /u/.

[9] Please refer to Tomabech&Tomita[1988a] for detail.

[10] Noisy either due to limitation of the speech recognition device or due to the noisy environment, or both.

5.2 A Scheme to Access contextual memory during parsing

Now, we look at our integration of contextual (thematic) memory activity with this unification-based syntax/semantics parsing. In essence, we perform spreading activation in memory every time a local semantic restriction test succeeds during the syntax/semantics unification. Our algorithm is as follows:

FOR each sentence in the speech input **DO**;

1. When unification of one feature-structure and another feature-structure succeeds (syntactically well-formed), and this unification accompanies the addition of one concept (semantic case-frame) to another concept as a part of the receiving concept's features (namely, succeeds in meeting case-frame slot filling restrictions), then:
2. Activate the concept that succeeded in the above unification and semantic test (receiving another concept as its feature[11]).
3. Activate the concepts that are abstractions of the activated concept in the memory-net.
4. If an activated concept is a thematic root concept, then send the thematic activation to the concepts that are thematic-children of the node.
5. When unifications build sentential case-frames, activate the sentential case-frame with the highest level of thematic-activation. Deactivate all other sentential case-frames and non-sentential case-frames. Perform upward activation as in 3 and thematic-activation (4) for the chosen sentential case-frame.

END FOR;

To clarify, we are assuming a frame-based semantic-net as a representation of domain knowledge which is organized by inheritance links and also by relation (feature) links that are mapped with syntactic feature-attributes at some level of abstraction. We are also using links that group thematically related concepts. This is attained by having some nodes characterized as thematic root nodes packaging the thematic children nodes. This packaging can be thematic as well as episodic.

As we stated in the description of the algorithm above, we have two kinds of activations: 1) unification triggered conceptual activations; and 2) thematic concept triggered thematic activations[12]. Both are guided spreading activations.

[11] This reception of another concept as a specific feature of the concept is equivalent to *'concept refinement'* that is central to parsers such as MOPTRANS (Lytinen[1984]), DMTRANS (Tomabechi[1987]) and DM-COMMAND (Tomabechi&Tomita[1988b]).

[12] The thematic activation is analogous to DMTRANS (Tomabechi[1987])'s 'C-MARKER' marker passing, except that DMTRANS uses lexical activation of contextual markers whereas we use unification-triggered activation of thematic root nodes as the source of thematic activations.

6 Generator and Synthesis Module

The generation module of our system is called GENKIT (Tomita&Nyberg[1988]) which was developed at CMT as a transportable natural language generation system. It uses the pseudo-unification grammar which is in essence equivalent to the parsing grammar that is utilized by our parser. GENKIT compiles the grammar into a sentence generation functions which are evaluated at run-time to produce natural language. Since a full report on this system is in print from the CMT, we will omit any further description of this module in this paper. The speech synthesis module we have adopted in our system is a commercial product called DECTALK produced by DEC. It receives a text input produced by GENKIT and synthesizes a human voice in English in a variety of ages, sexes and tones. Similar products have been introduced by several Japanese companies for synthesizing Japanese voice and we are currently working on using these systems for English to Japanese translation[13]. Since these products are commercial and are not produced by CMT, we will omit any discussion of these systems in this paper.

7 Conclusion

We have reported our progress in building a speech-to-speech translation system and introduced our experimental system SPEECHTRANS. SPEECHTRANS was demonstrated at the MT Conference[14] sponsored by Carnegie Mellon University in June, 1988. However, the introduction of prototype speech-to-speech translation systems should not leave the impression that a practical speech-to-speech translation system is around the corner. In natural language processing research, even with text inputs, we have issues that are yet to be solved, especially in the areas of pragmatics, learning and memory. Because of the added complexity due to the noisy speech input, attaining an acceptable quality in speech-to-speech translation is still difficult. Since our public demonstration of SPEECHTRANS, we have added schemes for contextual processing integrating a marker-passing algorithm with the unification-based syntax/semantics. This integration has shown encouraging results in narrowing down the ambiguity even with larger semantic domains and has been accepted by the researchers outside CMU as well[15]. Combined with increase in accuracy of speech recognition hardware (such as researches by CMU's speech recognition project), we hope to integrate full pragmatic processing to the speech translation system so that our system will evolve into a level that applications such as interpreting telephony may be considered possible.

[13] As a speech recognition hardware, we are currently connecting SPHINX (Lee[1988]) as a English front-end.

[14] Second International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages, June 12-14, 1988.

[15] Such as ICOT's JPSG based parsing systems (personal communication).

ACKNOWLEDGMENTS

The authors wish to thank Shuji Morii and Matsushita Research Institute for their contribution of the custom speech recognition hardware used by our project, and members of the Center for Machine Translation for fruitful discussions. Lori Levin and Eric Nyberg were especially helpful in preparing the final version of this paper.

APPENDIX: Implementation[16]

SPEECHTRANS is implemented on two HP Bobcat AI Workstations connected by a high-speed net. One controls the speech recognition hardware and the translation programs and other controls the DECTALK. The speech recognition algorithms are firmware written in the custom recognition hardware and the low-level control program for the hardware is written in 'C'. The top-level control program of the SPEECHTRANS is written in HP COMMONLISP and directly evaluates the object-code of the recognition hardware control programs and passes the result of recognition to the top-level COMMONLISP functions of the phoneme-based parser.

The phoneme-based parser is written in HP COMMONLISP augmented by full/pseudo-unification packages. The natural language generator (GENKIT) is also written in HP COMMONLISP which receives a functional-structure output and generates natural language which is sent via network to the second HP Workstation to be supplied to the speech synthesis module. The run-time parsing and generaton grammars are precompiled for run-time efficiency.

The speech synthesis module is a commercial product built by DEC called DECTALK. It is capable of producing different types of voices (female, male, young, old, etc.) and at varying pitches and can receive either phonemes or text inputs. Since the generator outputs the text output, the input to DECTALK is a text input and it produces the synthesized human voice.

References

[1] Kaplan, R. and Bresnan, J. (1982) 'Lexical Functional Grammar: A formal System for Grammatical Representation'. In *The Mental Representation of Grammatical Relations*, ed J. Bresnan. MIT Press.

[2] Lee, K. (1988) *Large-Vocaburaly Speaker-Independent Continuous Speech Recognition: The SPHINX System*. CMU-CS-88-148. Carnegie Mellon University.

[3] Lytinen S. (1984) *The organization of knowledge in a multi-lingual, integrated parser*. Ph.D. thesis Yale University.

[4] Morii, S., Niyada, K., Fujii, S. and Hoshimi, M. (1985) 'Large Vocabulary Speaker-independent Japanese Speech Recognition System.' In *Proceedings of ICASSP85*.

[5] Nyberg, E. (1988) *The FrameKit User's Guide Version 2.0*. CMU-CMT-88-107. Carnegie Mellon University.

[6] Pollard, C. and Sag, A. (1987) *An Information-based Syntax and Semantics*. Vol 1. CSLI.

[7] Poesio, M. and Rullent, C. (1987) 'Modified Caseframe Parsing for Speech Understanding Systems'. In *Proceedings of the IJCAI-87*.

[8] Saito, H. and Tomita, M. (1988a) 'Understanding Noisy Sentences by an LR Parser'. In *Denshi-Joho Tsushin Gakkai SP88-28*.

[9] Saito, H. and Tomita, M. (1988b) 'Parsing Noisy Sentences', In *Proceedings of the COLING-88*.

[10] Tomabechi, H. (1987) 'Direct Memory Access Translation'. In *Proceedings of the IJCAI-87*.

[11] Tomabechi, H. and Tomita, (1988a) 'The Integration of Unification-based Syntax/Semantics and Memory-based Pragmatics for Real-Time Understanding of Noisy Continuous Speech Input'. In *Proceedings of the AAAI-88*.

[12] Tomabechi, H. and Tomita, M. (1988b) 'Application of the Direct Memory Access paradigm to natural language interfaces to knowledge-based systems' In *Proceedings of the COLING-88*'.

[13] Tomabechi, H. Mitamura, T. and Tomita, M. (1988) 'Direct Memory Access Translation for Speech Input:' A Massively Parallel Network of Episodic/Thematic and Phonological Memory. In *Proceedings of the International Conference on Fifth Generation Computer Systems 1988* (FGCS'88).

[14] Tomita, M. (1985) *Efficient Parsing for Natural Language: A Fast algorithm for Practical Systems*. Kluwer Academic Publishers, Boston, MA.

[15] Tomita, M. (1986) 'An Efficient Word Lattice Parsing Algorithm for Continuous Speech Recognition'. In *Proceedings of ICASSP86*.

[16] Tomita, M. and Carbonell. J. (1987) 'The Universal Parser Architecture for Knowledge-Based Machine Translation'. In *Proceedings of IJCAI-87*.

[17] Tomita, M., Kee, M., Mitamura, T. and Carbonell, J. (1987) 'Linguistic and Domain Knowledge Sources for the Universal Parser Architecture'. In *Terminology and Knowledge Engineering* Eds. H. Czap, and C. Galinski INDEKS Verlag.

[18] Tomita, M. and Nyberg, E. (1988) *The Generation Kit and The Transformation Kit Version 3-2. Users Manual*. CMU-CMT-MEMO. Carnegie Mellon University.

[19] Wroblewski, D. (1987) 'Nondestructive graph unification'. In *Proceedings of AAAI-87*.

[16] The audio tape-recording of sample runs of our system is available from the authors by request.