RECENT ADVANCES IN JANUS: A SPEECH TRANSLATION SYSTEM

M.Woszczyna, N.Coccaro, A.Eisele, A.Lavie, A.McNair, T.Polzin, I.Rogina, C.P.Rose, T.Sloboda, M.Tomita, J.Tsutsumi, N.Aoki-Waibel, A.Waibel, W. Ward

Carnegie Mellon University
University of Karlsruhe

ABSTRACT

We present recent advances from our efforts in increasing coverage, robustness, generality and speed of JANUS, CMU's speech-to-speech translation system. JANUS is a speaker-independent system which translates spoken utterances in English and also in German into one of German, English or Japanese. The system has been designed around the task of conference registration (CR). It has initially been built based on a speech database of 12 read dialogs, encompassing a vocabulary of around 500 words. We have since been expanding the system along several dimensions to improve speed, robustness and coverage and to move toward spontaneous input.

1. INTRODUCTION

In this paper we describe recent improvements of JANUS, a speech to speech translation system. Improvements have been made mainly along the following dimensions: 1.) better context-dependent modeling improves performance in the speech recognition module, 2.) improved language models, smoothing, and word equivalence classes improve coverage and robustness of the sentences that the system accepts, 3.) an improved N-best search reduces run-time from several minutes to now real time, 4.) trigram and parser rescoring improves selection of suitable hypotheses from the N-best list for subsequent translation. On the machine translation side, 5.) a cleaner interlingua was designed and syntactic and domain-specific analysis were separated for greater reusability of components and greater quality of translation, 6.) a semantic parser was developed to achieve semantic analysis, should more careful analysis fail.

The JANUS [1] framework as it is presented here also allows us to experiment with components of a speech translation system, in an effort to achieve both robustness and high-quality translation. In the following we describe these efforts and system components that have been developed to date. At present, JANUS consists conceptually of three major components: speech recognition, machine translation and speech synthesis.

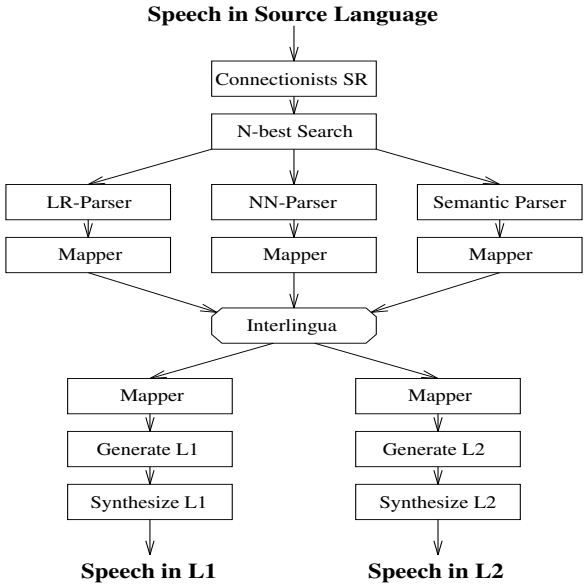

Figure 1: Overview of the System

2. RECOGNITION ENGINE

Our recognition engine uses several techniques to optimize the overall system performance. Speech input is preprocessed into time frames of spectral coefficients. Acoustic models are trained to give a score for each phoneme, representing the phoneme probability at the given frame. These scores are used by an N-best search algorithm to produce a list of sentence hypotheses. Based on this list, more computationally expensive language models are then applied to achieve further improvement of recognition accuracy.

2.1. Acoustic modeling

For acoustic modeling, several alternative algorithms are being evaluated including TDNN, MS-TDNN, MLP and LVQ [4, 3]. In the main JANUS system, an LVQ algorithm with context-dependent phonemes is now used for speaker independent recognition. For each phoneme, there is a context independent set of prototypical vectors. The output scores for each phoneme segment are computed from the euclidian distance using context de-

pendent segment weights.

Error rates using context dependent phonemes are lower by a factor 2 to 3 for English (1.5 to 2 for German) than using context independent phonemes. Results are shown in table 1.

	English		German	
language model	PP	WA	PP	WA
none	400.0	58.2	425.0	63.0
word-pairs	28.9	83.4	20.8	89.1
bigrams	16.2	92.6	18.3	93.7
smoothed bigrams	18.1	91.5	28.90	84.7
after resorting	—-	98.8		

Table 1: Word Accuracy for First Hypothesis

The performance on the RM-task at comparable perplexities is significantly better than for the CR-task, suggesting that the CR-task is somewhat more difficult.

2.2. Search

The search module of the recognizer builds a sorted list of sentence hypotheses. Speed and memory requirements have been dramatically improved: Though the amount of hypotheses computed for each utterance was increased from 6 to 100 hypotheses, the time required for their computation was reduced from typically 3 minutes to 3 seconds.

This was achieved by implementing the word dependent N-best algorithm[2] as a backward pass in the forward backward algorithm: First a fast firstbest search is performed, saving the scores at each possible word ending. In a second pass, this information is used for aggressive pruning to reduce the search effort for the N-best search. Further speedup was achieved by dynamically adapting the beam width to keep the number of active states constant, and by carefully avoiding the evaluation of states in large inactive regions of words. Important for total system performance is the fact that the firstbest hypothesis can already be analyzed by the MT modules while the N-best list is computed.

All language models (word-pairs, bigrams or smoothed bigrams, and trigrams for resorting) are now trained on more than 1000 CR-sentences, using word class specific equivalence classes (digits, names, towns, languages etc.)

2.3. Resorting

The resulting N-best list is resorted using trigrams to further improve results. Resorting improves the word accuracy for the best scoring hypothesis (created using smoothed bigrams) from 91.5% to 98%, and the average rank of the correct hypothesis within the list from 5.7 to 1.1; Much longer N-best lists have been used for experiments (500-1000). For practical application, a number of 100 hypotheses was found to be best.

3. THE MACHINE TRANSLATION (MT) ENGINE

The MT-component that we have previously used has now been replaced by a new module that can run several alternate processing strategies in parallel. In translating spoken language from one language to another, the analysis of spoken sentences which suffer from ill-formed input and recognition errors is most certainly the hardest part. Based on the list of N-best hypotheses delivered by the recognition engine, we can now attempt to select and analyze the most plausible sentence hypothesis in view of producing an accurate and meaningful translation. Two goals are central in this attempt: *high fidelity* and *accurate translation* wherever possible, and *robustness* or *graceful degradation*, should attempts for high fidelity translation fail in face of ill-formed or misrecognized input. At present, three parallel modules attempt to address these goals: 1) an LR-parser based syntactic approach, 2) a semantic pattern based approach and 3) a connectionist approach. The most useful analysis from these modules is mapped onto a common Interlingua, a language independent, but domain-specific representation of meaning. The parsing stage attempts to derive a high precision analysis first, using a strict syntax and domain specific semantics. Connectionist and/or semantic parsers are currently applied as back-up, if the higher precision analysis fails. The Interlingua ensures that alternate modules can be applied in a modular fashion and that different output languages can be added without redesign of the analysis stage.

3.1. Generalized LR Parser

The first step of the translation process is syntactic parsing with the Generalized LR Parser/Compiler. The Generalized LR parsing algorithm is an extension of LR parsing with the special device called "Graph-Structured Stack" [9], and it can handle arbitrary context-free grammars while most of the LR efficiency is preserved. A grammar with about 455 rules for general colloquial English is written in a Pseudo Unification formalism, that is similar to Unification Grammar and LFG formalisms.

Robust GLR Parsing: In case the standard parsing procedure fails to parse an input sentence, the robust version of the parser nondeterministically skips one or more words in the sentence, and returns the parse with the fewest skipped words. In this mode, the parser will return some parse(s) with any input sentence, unless no part of the sentence can be interpreted at all.

When the standard GLR parser fails on all sentence candidates, this robust GLR parser is applied to the best sentence candidate.

3.2. The Interlingua

The output of the parser, known as "syntactic f-structure", is then fed into a mapper to produce an Interlingua representation. For the mapper, we use a software tool known as Transformation Kit [10]. A mapping grammar with about 300 rules is written for the Conference Registration domain of English.

```
((PREV-UTTERANCES ((SPEECH-ACT *ACKNOWL) (VALUE *HELLO)))
  (TIME *PRESENT)
  (PARTY
    ((DEFINITE +) (NUMBER *SG)
              (ANIM -)
              (TYPE *CONFERENCE)
              (CONCEPT *OFFICE)))
  (SPEECH-ACT *IDENTIFY-OTHER))
```

Figure 2: Example: Interlingua Output

Figure 2 is an example of Interlingua representation produced from the sentence "Hello is this the conference office". In the example, "Hello" is represented as speech-act *ACKNOWLEDGEMENT, and the rest as speech-act *IDENTFY-OTHER.

3.3. The Generator

The generation of target language from an Interlingua representation involves two steps. First, with the same Transformation Kit used in the analysis phase, Interlingua representation is mapped into syntactic f-structure of the target language. There are about 300 rules in the generation mapping grammar for German, and 230 rules for Japanese. The f-structure is then fed into sentence generation software called "GENKIT" [10] to produce a sentence in the target language. A grammar for GENKIT is written in the same formalism as the Generalized LR Parser: phrase structure rules augmented with pseudo unification equations. The GENKIT grammar for general colloquial German has about 90 rules, and Japanese about 60 rules. Software called MORPHE is also used for morphlogical generation of German.

3.4. Semantic Pattern Based Parsing

Our robust semantic parser combines frame based semantics with semantic phrase grammars. We use a frame based parser similar to the DYPAR parser used by Carbonell, et al. to process ill-formed text[6], and the MINDS system previously developed at CMU. Semantic information is represented in a set of frames. Each frame contains a set of slots representing pieces of information. In order to fill the slots in the frames, we use semantic fragment grammars. The operation of the parser can be viewed as "phrase spotting". A beam of possible interpretations are pursued simultaneously. An interpretation is a frame with some of its slots filled. The RTNs perform pattern matches against the input string. When a phrase is recognized, it attempts to extend all current interpretations. That is, it is assigned to slots in active interpretations that it can fill. Phrases assigned to slots in the same interpretation are not allowed to overlap. In case of overlap, multiple interpretations are produced. When two interpretations for the same frame end with the same phrase, the lower scoring one is pruned. This amounts to dynamic programming on series of phrases. The score for an interpretation is the number of input words that it accounts for.

Each slot type is represented by a separate Recursive Transition Network, which specifies all ways of saying the meaning represented by the slot. The grammar is a semantic grammar, non-terminals are semantic concepts instead of parts of speech. The grammar is also written so that information carrying fragments (semantic fragments) can stand alone (be recognized by a net) as well as being embedded in a sentence. Fragments which do not form a grammatical English sentence are still parsed by the system. Here there is not one large network representing all sentence level patterns, but many small nets representing information carrying chunks. Networks can "call" other networks, thereby significantly reducing the overall size of the system. These networks are used to perform pattern matches against input word strings.

The parsing grammar was designed so that each frame has exactly one corresponding speech act. Each top level slot corresponds to some thematic role or other major semantic concept such as action. Subnets correspond to more specific semantic classes of constituents. In this way, the interpretation returned by the parser can be easily mapped onto the interlingua and missing information can be filled by meaningful default values with minimal effort.

3.5. Connectionist Parsing

The connectionist parsing system PARSEC [7] is used as a fall-back module if the symbolic high precision one fails to analyze the input. The important aspect of the PARSEC system is that it learns to parse sentences from a corpus of training examples.

Because PARSEC learns and generalizes from the examples given in the training set no explicit grammar rules have to be specified by hand. In particular, this is of importance when the system has to cope with spontaneous utterances which frequently are "corrupted" with disfluencies, restarts, repairs or ungrammatical constructions. To specify symbolic grammars capturing these phenomena has been proven to be very difficult. On the other

side there is a "built-in" robustness against these phenomena in a connectionist system.

The connectionist parsing process is able to combine symbolic information (e.g. syntactic features of words) with non-symbolic information (e.g. statistical likelihood of sentence types). Moreover, the system can easily integrate different knowledge sources. For example, instead of just training on the symbolic input string we trained PARSEC on both the symbolic input string and the pitch contour. After training was completed the system was able to use the additional information to determine the sentence mood in cases where syntactic clues were not sufficient. We are considering to extend the idea of integrating prosodic information into the parsing process in order to increase the performance of the system when it is confronted with corrupted input. We hope that prosodic information will help to indicate restarts and repairs.

The current PARSEC system comprises six hierarchically ordered (back-propagation) connectionist modules. Each module is responsible for a specific task. For example, there are two modules which determine phrase and clause boundaries. Other modules are responsible for assigning to phrases or clauses labels which indicate their function and/or relationship to other constituents. The top module determines the mood of the sentence.

4. SYSTEM INTEGRATION

The system accepts continuous speech speaker–independently in either input language, and produces synthetic speech output in near real-time. Our system can be linked to different language versions of the system or corresponding partner systems via ethernet or via telephone modem lines. This possibility has recently been tested between sites in the US, Japan and Germany to illustrate the possibility of international telephone speech translation.

The minimal equipment for this system is a Gradient Desklab 14 A/D-converter, an HP 9000/730 (64 Meg RAM) workstation for each input laguage, and a DECtalk speech synthesizer.

For our current system, we have eliminated considerable amounts of communication delays by introducing socket communication between pipelined parts of the system. Thus the search can start before the preprocessing program is done, and the parser starts working on the first hypothesis while the N-best list is computed.

5. CONCLUSION

In this paper, we have discussed recent extensions to the JANUS system a speaker independent multi-lingual speech-to-speech translation system under development at Carnegie Mellon and Karlsruhe University. The components include a speech recognizer using an N-best sentence search, to derive alternate hypotheses for later processing during the translation. The MT component attempts to produce a high-accuracy translation using precise syntactic and semantic analysis. Should this analysis fail due to ill-formed input or misrecognitions, a connectionist parser, PARSEC, and a semantic parser produce alternate minimalist analyses, to at least establish the basic meaning of an input utterance. Human-to-human dialogs appear to generate a larger and more varied breadth of expression than human-machine dialogs. Further research is in progress to quantify this observation and to increase robustness and coverage of the system in this environment.

References

1. L. Osterholtz, A. McNair, I. Rogina, H. Saito, T. Sloboda, J. Tebelskis, A. Waibel, and M. Woszczyna. *Testing Generality in JANUS: A Multi-Lingual Speech to Speech Translation System*, volume 1, pp 209–212. ICASSP 1992.
2. Austin S., Schwartz R. *A Comparison of Several Approximate Algorithms for Finding N-best Hypotheses*, ICASSP 1991, volume 1, pp 701–704.
3. O. Schmidbauer and J. Tebelskis. *An LVQ based Reference Model for Speaker-Adaptive Speech Recognition.* ICASSP 1992, volume 1, pages 441–444.
4. J. Tebelskis and A. Waibel. *Performance through consistency: MS-TDNNs for large vocabulary continuous speech recognition*, Advances in Neural Information Processing Systems, Morgan Kaufmann.
5. W. Ward, *Understanding Spontaneous Speech*, DARPA Speech and Natural Language Workshop 1989, pp 137–141.
6. J.G. Carbonell and P.J. Hayes, *Recovery Strategies for Parsing Extragrammatical Language*, Carnegie-Mellon University Computer Science Technical Report 1984, (CMU-CS-84-107)
7. A.J. Jain, A. Waibel, D. Touretzky, *PARSEC: A Structured Connectionist Parsing System for Spoken Language*, ICASSP 1992, volume 1, pp 205–208.
8. T.S. Polzin, *Pronoun Resolution. Interaction of Syntactic and Semantic Information in Connectionist Parsing*, Thesis, Carnegie Mellon University, Department of Philosophy, Computational Linguistics, in preparation.
9. Tomita, M. (ed.), *Generalized LR Parsing*, Kluwer Academic Publishers, Boston MA, 1991.
10. Tomita, M. and Nyberg, E.; *The Generation Kit and The Transformation Kit: User's Guide* Technical Memo, Center for Machine Translation, Carnegie Mellon University, CMU-CMT-88-MEMO, 1988
11. Lavie, A and Tomita, M.; *An Efficient Word-Skipping Parsing Algorithm for Context-Free Grammars* submitted to 3rd International Workshop on Parsing Technologies (IWPT93) Belguim, 1993.