Proceedings of the Workshop on Speech-to-Speech Translation:
Algorithms and Systems, Philadelphia, July 2002, pp. 15-22.
Association for Computational Linguistics.

Spoken Language Parsing Using Phrase-Level Grammars and Trainable Classifiers

Chad Langley, **Alon Lavie**, **Lori Levin**, **Dorcas Wallace**, **Donna Gates**, and **Kay Peterson**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, USA
{clangley|alavie|lsl|dorcas|dmg|kay}@cs.cmu.edu

Abstract

In this paper, we describe a novel approach to spoken language analysis for translation, which uses a combination of grammar-based phrase-level parsing and automatic classification. The job of the analyzer is to produce a shallow semantic interlingua representation for spoken task-oriented utterances. The goal of our hybrid approach is to provide accurate real-time analyses while improving robustness and portability to new domains and languages.

1 Introduction

Interlingua-based approaches to Machine Translation (MT) are highly attractive in systems that support a large number of languages. For each source language, an analyzer that converts the source language into the interlingua is required. For each target language, a generator that converts the interlingua into the target language is needed. Given analyzers and generators for all supported languages, the system simply connects the source language analyzer with the target language generator to perform translation.

Robust and accurate analysis is critical in interlingua-based translation systems. In speech-to-speech translation systems, the analyzer must be robust to speech recognition errors, spontaneous speech, and ungrammatical inputs as described by Lavie (1996). Furthermore, the analyzer should run in (near) real time.

In addition to accuracy, speed, and robustness, the portability of the analyzer with respect to new domains and new languages is an important consideration. Despite continuing improvements in speech recognition and translation technologies, restricted domains of coverage are still necessary in order to achieve reasonably accurate machine translation. Porting translation systems to new domains or even expanding the coverage in an existing domain can be very difficult and time-consuming. This creates significant challenges in situations where translation is needed for a new domain within relatively short notice. Likewise, demand can be high for translation systems that can be rapidly expanded to include new languages that were not previously considered important. Thus, it is important that the analysis approach used in a translation system be portable to new domains and languages.

One approach to analysis in restricted domains is to use semantic grammars, which focus on parsing semantic concepts rather than syntactic structure. Semantic grammars can be especially useful for parsing spoken language because they are less susceptible to syntactic deviations caused by spontaneous speech effects. However, the focus on meaning rather than syntactic structure generally makes porting to a new domain quite difficult. Since semantic grammars do not exploit syntactic similarities across domains, completely new grammars must usually be developed.

While grammar-based parsing can provide very accurate analyses on development data, it is difficult for a grammar to completely cover a domain, a problem that is exacerbated by spoken input. Furthermore, it generally takes a great deal of effort by human experts to develop a high-coverage grammar. On the other hand, machine learning approaches can generalize beyond training data and tend to degrade gracefully in the face of noisy input. Machine learning methods may, however, be less accurate on clearly in-domain input than grammars and may require a large amount of training data.

We describe a prototype version of an analyzer that combines phrase-level parsing and machine

learning techniques to take advantage of the benefits of each. Phrase-level semantic grammars and a robust parser are used to extract low-level interlingua arguments from an utterance. Then, automatic classifiers assign high-level domain actions to semantic segments in the utterance.

2 MT System Overview

The analyzer we describe is used for English and German in several multilingual human-to-human speech-to-speech translation systems, including the NESPOLE! system (Lavie et al., 2002). The goal of NESPOLE! is to provide translation for common users within real-world e-commerce applications. The system currently provides translation in the travel and tourism domain between English, French, German and Italian.

NESPOLE! employs an interlingua-based translation approach that uses four basic steps to perform translation. First, an automatic speech recognizer processes spoken input. The best-ranked hypothesis from speech recognition is then passed through the analyzer to produce interlingua. Target language text is then generated from the interlingua. Finally, the target language text is synthesized into speech.

This interlingua-based translation approach allows for distributed development of the components for each language. The components for each language are assembled into a translation server that accepts speech, text, or interlingua as input and produces interlingua, text, and synthesized speech. In addition to the analyzer described here, the English translation server uses the JANUS Recognition Toolkit for speech recognition, the GenKit system (Tomita & Nyberg, 1988) for generation, and the Festival system (Black et al., 1999) for synthesis.

NESPOLE! uses a client-server architecture (Lavie et al., 2001) to enable users who are browsing the web pages of a service provider (e.g. a tourism bureau) to seamlessly connect to a human agent who speaks a different language. Using commercially available software such as Microsoft NetMeeting™, a user is connected to the NESPOLE! Mediator, which establishes connections with the agent and with translation servers for the appropriate languages. During a dialogue, the Mediator transmits spoken input from the users to the translation servers and synthesized translations from the servers to the users.

3 The Interlingua

The interlingua used in the NESPOLE! system is called Interchange Format (IF) (Levin et al., 1998; Levin et al., 2000). The IF defines a shallow semantic representation for task-oriented utterances that abstracts away from language-specific syntax and idiosyncrasies while capturing the meaning of the input. Each utterance is divided into semantic segments called *semantic dialog units* (SDUs), and an IF is assigned to each SDU. An IF representation consists of four parts: a speaker tag, a speech act, an optional sequence of concepts, and an optional set of arguments. The representation takes the following form:

```
speaker : speech act +concept* (argument*)
```

The speaker tag indicates the role of the speaker in the dialogue. The speech act captures the speaker's intention. The concept sequence, which may contain zero or more concepts, captures the focus of an SDU. The speech act and concept sequence are collectively referred to as the domain action (DA). The arguments use a feature-value representation to encode specific information from the utterance. Argument values can be atomic or complex. The IF specification defines all of the components and describes how they can be legally combined. Several examples of utterances with corresponding IFs are shown below.

Thank you very much.

```
a:thank
```

Hello.

```
c:greeting (greeting=hello)
```

How far in advance do I need to book a room for the Al-Cervo Hotel?

```
c:request-suggestion+reservation+room (
 suggest-strength=strong,
 time=(time-relation=before,
  time-distance=question),
 who=i,
 room-spec=(room, identifiability=no,
  location=(object-name=cervo_hotel)))
```

4 The Hybrid Analysis Approach

Our hybrid analysis approach uses a combination of grammar-based parsing and machine learning techniques to transform spoken utterances into the IF representation described above. The speaker tag is assumed to be given. Thus, the goal of the analyzer is to identify the DA and arguments.

The hybrid analyzer operates in three stages. First, semantic grammars are used to parse an

utterance into a sequence of arguments. Next, the utterance is segmented into SDUs. Finally, the DA is identified using automatic classifiers.

4.1 Argument Parsing

The first stage in analysis is parsing an utterance for arguments. During this stage, utterances are parsed with phrase-level semantic grammars using the robust SOUP parser (Gavaldà, 2000).

4.1.1 The Parser

The SOUP parser is a stochastic, chart-based, top-down parser that is designed to provide real-time analysis of spoken language using context-free semantic grammars. One important feature provided by SOUP is word skipping. The amount of skipping allowed is configurable and a list of unskippable words can be defined. Another feature that is critical for phrase-level argument parsing is the ability to produce analyses consisting of multiple parse trees. SOUP also supports modular grammar development (Woszczyna et al., 1998). Subgrammars designed for different domains or purposes can be developed independently and applied in parallel during parsing. Parse tree nodes are then marked with a subgrammar label. When an input can be parsed in multiple ways, SOUP can provide a ranked list of interpretations.

In the prototype analyzer, word skipping is only allowed between parse trees. Only the best-ranked argument parse is used for further processing.

4.1.2 The Grammars

Four grammars are defined for argument parsing: an argument grammar, a pseudo-argument grammar, a cross-domain grammar, and a shared grammar. The argument grammar contains phrase-level rules for parsing arguments defined in the IF. Top-level argument grammar nonterminals correspond to top-level arguments in the IF.

The pseudo-argument grammar contains top-level nonterminals that do not correspond to interlingua concepts. These rules are used for parsing common phrases that can be grouped into classes to capture more useful information for the classifiers. For example, *all booked up*, *full*, and *sold out* might be grouped into a class of phrases that indicate unavailability. In addition, rules in the pseudo-argument grammar can be used for contextual anchoring of ambiguous arguments. For example, the arguments [who=] and [to-whom=] have the same values. To parse these arguments properly in a sentence like "*Can you send me the brochure?*", we use a pseudo-argument grammar rule, which refers to the arguments [who=] and [to-whom=] within the appropriate context.

The cross-domain grammar contains rules for parsing whole DAs that are domain-independent. For example, this grammar contains rules for greetings (*Hello*, *Good bye*, *Nice to meet you*, etc.). Cross-domain grammar rules do not cover all possible domain-independent DAs. Instead, the rules focus on DAs with simple or no argument lists. Domain-independent DAs with complex argument lists are left to the classifiers. Cross-domain rules play an important role in the prediction of SDU boundaries.

Finally, the shared grammar contains common grammar rules that can be used by all other subgrammars. These include definitions for most of the arguments, since many can also appear as sub-arguments. RHSs in the argument grammar contain mostly references to rules in the shared grammar. This method eliminates redundant rules in the argument and shared grammars and allows for more accurate grammar maintenance.

4.2 Segmentation

The second stage of processing in the hybrid analysis approach is segmentation of the input into SDUs. The IF representation assigns DAs at the SDU level. However, since dialogue utterances often consist of multiple SDUs, utterances must be segmented into SDUs before DAs can be assigned. Figure 1 shows an example utterance containing four arguments segmented into two SDUs.

SDU1		**SDU2**	
greeting=	**disposition=**	**visit-spec=**	**location=**
hello	*i would like to*	*take a vacation*	*in val di fiemme*

Figure 1. Segmentation of an utterance into SDUs.

The argument parse may contain trees for cross-domain DAs, which by definition cover a complete SDU. Thus, there must be an SDU boundary on both sides of a cross-domain tree. Additionally, no SDU boundaries are allowed within parse trees. The prototype analyzer drops words skipped between parse trees, leaving only a sequence of trees. The parse trees on each side of a potential boundary are examined, and if either tree was constructed by the cross-domain grammar, an SDU boundary is inserted. Otherwise, a simple statistical

model similar to the one described by Lavie et al. (1997) estimates the likelihood of a boundary.

The statistical model is based only on the root labels of the parse trees immediately preceding and following the potential boundary position. Suppose the position under consideration looks like $[A_1 \bullet A_2]$, where there may be a boundary between arguments A_1 and A_2. The likelihood of an SDU boundary is estimated using the following formula:

$$F([A_1 \bullet A_2]) \approx \frac{C([A_1\bullet]) + C([\bullet A_2])}{C([A_1]) + C([A_2])}$$

The counts $C([A_1\bullet])$, $C([\bullet A_2])$, $C([A_1])$, $C([A_2])$ are computed from the training data. An evaluation of this baseline model is presented in section 6.

4.3 DA Classification

The third stage of analysis is the identification of the DA for each SDU using automatic classifiers. After segmentation, a cross-domain parse tree may cover an SDU. In this case, analysis is complete since the parse tree contains the DA. Otherwise, automatic classifiers are used to assign the DA. In the prototype analyzer, the DA classification task is split into separate subtasks of classifying the speech act and concept sequence. This reduces the complexity of each subtask and allows for the application of specialized techniques to identify each component.

One classifier is used to identify the speech act, and a second classifier identifies the concept sequence. Both classifiers are implemented using TiMBL (Daelemans et al., 2000), a memory-based learner. Speech act classification is performed first. Input to the speech act classifier is a set of binary features that indicate whether each of the possible argument and pseudo-argument labels is present in the argument parse for the SDU. No other features are currently used. Concept sequence classification is performed after speech act classification. The concept sequence classifier uses the same feature set as the speech act classifier with one additional feature: the speech act assigned by the speech act classifier. We present an evaluation of this baseline DA classification scheme in section 6.

4.4 Using the IF Specification

The IF specification imposes constraints on how elements of the IF representation can legally combine. DA classification can be augmented with knowledge of constraints from the IF specification, providing two advantages over otherwise naïve classification. First, the analyzer must produce valid IF representations in order to be useful in a translation system. Second, using knowledge from the IF specification can improve the quality of the IF produced, and thus the translation.

Two elements of the IF specification are especially relevant to DA classification. First, the specification defines constraints on the composition of DAs. There are constraints on how concepts are allowed to pair with speech acts as well as ordering constraints on how concepts are allowed to combine to form a valid concept sequence. These constraints can be used to eliminate illegal DAs during classification. The second important element of the IF specification is the definition of how arguments are licensed by speech acts and concepts. In order for an IF to be valid, at least one speech act or concept in the DA must license each argument.

The prototype analyzer uses the IF specification to aid classification and guarantee that a valid IF representation is produced. The speech act and concept sequence classifiers each provide a ranked list of possible classifications. When the best speech act and concept sequence combine to form an illegal DA or form a legal DA that does not license all of the arguments, the analyzer attempts to find the next best legal DA that licenses the most arguments. Each of the alternative concept sequences (in ranked order) is combined with each of the alternative speech acts (in ranked order). For each possible legal DA, the analyzer checks if all of the arguments found during parsing are licensed. If a legal DA is found that licenses all of the arguments, then the process stops. If not, one additional fallback strategy is used. The analyzer then tries to combine the best classified speech act with each of the concept sequences that occurred in the training data, sorted by their frequency of occurrence. Again, the analyzer checks if each legal DA licenses all of the arguments and stops if such a DA is found. If this step fails to produce a legal DA that licenses all of the arguments, the best-ranked DA that licenses the most arguments is returned. In this case, any arguments that are not licensed by the selected DA are removed. This approach is used because it is generally better to select an alternative DA and retain more arguments

than to keep the best DA and lose the information represented by the arguments. An evaluation of this strategy is presented in the section 6.

5 Grammar Development and Classifier Training

During grammar development, it is generally useful to see how changes to the grammar affect the IF representations produced by the analyzer. In a purely grammar-based analysis approach, full interlingua representations are produced as the result of parsing, so testing new grammars simply requires loading them into the parser. Because the grammars used in our hybrid approach parse at the argument level, testing grammar modifications at the complete IF level requires retraining the segmentation model and the DA classifiers.

When new grammars are ready for testing, utterance-IF pairs for the appropriate language are extracted from the training database. Each utterance-IF pair in the training data consists of a single SDU with a manually annotated IF. Using the new grammars, the argument parser is applied to each utterance to produce an argument parse. The counts used by the segmentation model are then recomputed based on the new argument parses. Since each utterance contains a single SDU, the counts $C([\bullet A_2])$ and $C([A_1 \bullet])$ can be computed directly from the first and last arguments in the parse respectively.

Next, the training examples for the DA classifiers are constructed. Each training example for the speech act classifier consists of the speech act from the annotated IF and a vector of binary features with a positive value set for each argument or pseudo-argument label that occurs in the argument parse. The training examples for the concept sequence classifiers are similar with the addition of the annotated speech act to the feature vector. After the training examples are constructed, new classifiers are trained.

Two tools are available to support easy testing during grammar development. First, the entire training process can be run using a single script. Retraining for a new grammar simply requires running the script with pointers to the new grammars. Then, a special development mode of the translation servers allows the grammar writers to load development grammars and their corresponding segmentation model and DA classifiers. The translation server supports input in the form of individual utterances or files and allows the grammar developers to look at the results of each stage of the analysis process.

6 Evaluation

We present the results from recent experiments to measure the performance of the analyzer components and of end-to-end translation using the analyzer. We also report the results of an ablation experiment that used earlier versions of the analyzer and IF specification.

6.1 Translation Experiment

	Acceptable	Perfect
SR Hypotheses	66%	56%
Translation from Transcribed Text	58%	43%
Translation from SR Hypotheses	45%	32%

Table 1. English-to-English end-to-end translation

	Acceptable	Perfect
Translation from Transcribed Text	55%	38%
Translation from SR Hypotheses	43%	27%

Table 2. English-to-Italian end-to-end translation

Tables 1 and 2 show end-to-end translation results of the NESPOLE! system. In this experiment, the input was a set of English utterances. The utterances were paraphrased back into English via the interlingua (Table 1) and translated into Italian (Table 2). The data used to train the DA classifiers consisted of 3350 SDUs annotated with IF representations. The test set contained 151 utterances consisting of 332 SDUs from 4 unseen dialogues. Translations were compared to human transcriptions and graded as described in (Levin et al., 2000). A grade of `perfect`, `ok`, or `bad` was assigned to each translation by human graders. A grade of `perfect` or `ok` is considered acceptable. The table shows the average of grades assigned by three graders.

The row in Table 1 labeled *SR Hypotheses* shows the grades when the speech recognizer output is compared directly to human transcripts. As these grades show, recognition errors can be a

major source of unacceptable translations. These grades provide a rough bound on the translation performance that can be expected when using input from the speech recognizer since meaning lost due to recognition errors cannot be recovered. The rows labeled *Translation from Transcribed Text* show the results when human transcripts are used as input. These grades reflect the combined performance of the analyzer and generator. The rows labeled *Translation from SR Hypotheses* show the results when the speech recognizer produces the input utterances. As expected, translation performance was worse with the introduction of recognition errors.

Precision	Recall
70%	54%

Table 3. SDU boundary detection performance

Table 3 shows the performance of the segmentation model on the test set. The SDU boundary positions assigned automatically were compared with manually annotated positions.

	Classifier Accuracy
Speech Act	65%
Concept Sequence	54%
Domain Action	43%

Table 4. Classifier accuracy on transcription

	Frequency
Speech Act	33%
Concept Sequence	40%
Domain Action	14%

Table 5. Frequency of most common DA elements

Table 4 shows the performance of the DA classifiers, and Table 5 shows the frequency of the most common DA, speech act, and concept sequence in the test set. Transcribed utterances were used as input and were segmented into SDUs before analysis. This experiment is based on only 293 SDUs. For the remaining SDUs in the test set, it was not possible to assign a valid representation based on the current IF specification.

These results demonstrate that it is not always necessary to find the canonical DA to produce an acceptable translation. This can be seen by comparing the *Domain Action* accuracy from Table 4 with the *Transcribed* grades from Table 1. Although the DA classifiers produced the canonical DA only 43% of the time, 58% of the translations were graded as acceptable.

	Changed
Speech Act	5%
Concept Sequence	26%
Domain Action	29%

Table 6. DA elements changed by IF specification

In order to examine the effects of using IF specification constraints, we looked at the 182 SDUs which were not parsed by the cross-domain grammar and thus required DA classification. Table 6 shows how many DAs, speech acts, and concept sequences were changed as a result of using the constraints. DAs were changed either because the DA was illegal or because the DA did not license some of the arguments. Without the IF specification, 4% of the SDUs would have been assigned an illegal DA, and 29% of the SDUs (those with a changed DA) would have been assigned an illegal IF. Furthermore, without the IF specification, 0.38 arguments per SDU would have to be dropped while only 0.07 arguments per SDU were dropped when using the fallback strategy. The mean number of arguments per SDU was 1.47.

6.2 Ablation Experiment

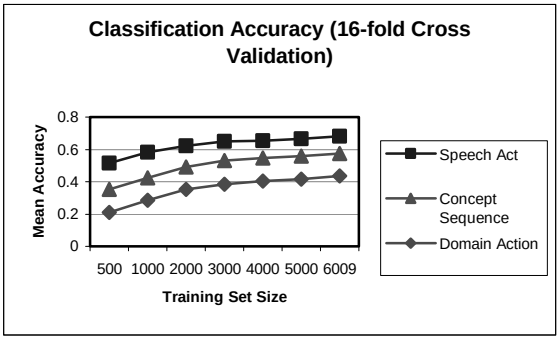

Figure 2: DA classifier accuracy with varying amounts of data

Figure 2 shows the results of an ablation experiment that examined the effect of varying the training set size on DA classification accuracy. Each point represents the average accuracy using a 16-fold cross validation setup.

The training data contained 6409 SDU-interlingua pairs. The data were randomly divided

into 16 test sets containing 400 examples each. In each fold, the remaining data were used to create training sets containing 500, 1000, 2000, 3000, 4000, 5000, and 6009 examples.

The performance of the classifiers appears to begin leveling off around 4000 training examples. These results seem promising with regard to the portability of the DA classifiers since a data set of this size could be constructed in a few weeks.

7 Related Work

Lavie et al. (1997) developed a method for identifying SDU boundaries in a speech-to-speech translation system. Identifying SDU boundaries is also similar to sentence boundary detection. Stevenson and Gaizauskas (2000) use TiMBL (Daelemans et al., 2000) to identify sentence boundaries in speech recognizer output, and Gotoh and Renals (2000) use a statistical approach to identify sentence boundaries in automatic speech recognition transcripts of broadcast speech.

Munk (1999) attempted to combine grammars and machine learning for DA classification. In Munk's SALT system, a two-layer HMM was used to segment and label arguments and speech acts. A neural network identified the concept sequences. Finally, semantic grammars were used to parse each argument segment. One problem with SALT was that the segmentation was often inaccurate and resulted in bad parses. Also, SALT did not use a cross-domain grammar or interlingua specification.

Cattoni et al. (2001) apply statistical language models to DA classification. A word bigram model is trained for each DA in the training data. To label an utterance, the most likely DA is assigned. Arguments are identified using recursive transition networks. IF specification constraints are used to find the most likely valid DA and arguments.

8 Discussion and Future Work

One of the primary motivations for developing the hybrid analysis approach described here is to improve the portability of the analyzer to new domains and languages. We expect that moving from a purely grammar-based parsing approach to this hybrid approach will help attain this goal.

The SOUP parser supports portability to new domains by allowing separate grammar modules for each domain and a grammar of rules shared across domains (Woszczyna et al., 1998). This modular grammar design provides an effective method for adding new domains to existing grammars. Nevertheless, developing a full semantic grammar for a new domain requires significant effort by expert grammar writers.

The hybrid approach reduces the manual labor required to port to new domains by incorporating machine learning. The most labor-intensive part of developing full semantic grammars for producing IF is writing DA-level rules. This is exactly the work eliminated by using automatic DA classifiers. Furthermore, the phrase-level argument grammars used in the analyzer contain fewer rules than a full semantic grammar. The argument-level grammars are also less domain-dependent than the full grammars and thus more reusable. The DA classifiers should also be more tolerant than full grammars of deviations from the domain.

We analyzed the grammars from a previous version of the translation system, which produced complete IFs using strictly grammar-based parsing, to estimate what portion of the grammar was devoted to the identification of domain actions. Approximately 2200 rules were used to cover 400 DAs. Nonlexical rules made up about half of the grammar, and the DA rules accounted for about 20% of the nonlexical rules. Using these figures, we can project the number of DA rules that would have to be added to the current system, which uses our hybrid analysis approach. The database for the new system contains approximately 600 DAs. Assuming the average number of rules per DA is the same as before, roughly 3300 DA-level rules would have to be added to the current grammar, which has about 17500 nonlexical rules, to cover the DAs in the database.

Our hybrid approach should also improve the portability of the analyzer to new languages. Since grammars are language specific, adding a new language still requires writing new argument grammars. Then the DA classifiers simply need to be retrained on data for the new language. If training data for the new language were not available, DA classifiers using only language-independent features, from the IF for example, could be trained on data for existing languages and used for the new language. Such classifiers could be used as a starting point until training data was available in the new language.

The experimental results indicate the promise of the analysis approach we have described. The

level of performance reported here was achieved using a simple segmentation model and simple DA classifiers with limited feature sets. We expect that performance will substantially improve with a more informed design of the segmentation model and DA classifiers. We plan to examine various design options, including richer feature sets and alternative classification techniques. We are also planning experiments to evaluate robustness and portability when the coverage of the NESPOLE! system is expanded to the medical domain later this year. In these experiments, we will measure the effort needed to write new argument grammars, the extent to which existing argument grammars are reusable, and the effort required to expand the argument grammar to include DA-level rules.

9 Acknowledgements

The research work reported here was supported by the National Science Foundation under Grant number 9982227. Special thanks to Alex Waibel and everyone in the NESPOLE! group for their support on this work.

References

Black, A., P. Taylor, and R. Caley. 1999. The Festival Speech Synthesis System: System Documentation. Human Computer Research Centre, University of Edinburgh, Scotland. http://www.cstr.ed.ac.uk/projects/festival/manual

Cattoni, R., M. Federico, and A. Lavie. 2001. Robust Analysis of Spoken Input Combining Statistical and Knowledge-Based Information Sources. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, Trento, Italy.

Daelemans, W., J. Zavrel, K. van der Sloot, and A. van den Bosch. 2000. TiMBL: Tilburg Memory Based Learner, version 3.0, Reference Guide. *ILK Technical Report 00-01*. http://ilk.kub.nl/~ilk/papers/ilk0001.ps.gz

Gavaldà, M. 2000. SOUP: A Parser for Real-World Spontaneous Speech. In *Proceedings of the IWPT-2000*, Trento, Italy.

Gotoh, Y. and S. Renals. Sentence Boundary Detection in Broadcast Speech Transcripts. 2000. In *Proceedings on the International Speech Communication Association Workshop: Automatic Speech Recognition: Challenges for the New Millennium*, Paris.

Lavie, A., F. Metze, F. Pianesi, et al. 2002. Enhancing the Usability and Performance of NESPOLE! – a Real-World Speech-to-Speech Translation System. In *Proceedings of HLT-2002*, San Diego, CA.

Lavie, A., C. Langley, A. Waibel, et al. 2001. Architecture and Design Considerations in NESPOLE!: a Speech Translation System for E-commerce Applications. In *Proceedings of HLT-2001*, San Diego, CA.

Lavie, A., D. Gates, N. Coccaro, and L. Levin. 1997. Input Segmentation of Spontaneous Speech in JANUS: a Speech-to-speech Translation System. In *Dialogue Processing in Spoken Language Systems: Revised Papers from ECAI-96 Workshop*, E. Maier, M. Mast, and S. Luperfoy (eds.), LNCS series, Springer Verlag.

Lavie, A. 1996. GLR*: A Robust Grammar-Focused Parser for Spontaneously Spoken Language. PhD dissertation, *Technical Report CMU-CS-96-126*, Carnegie Mellon University, Pittsburgh, PA.

Levin, L., D. Gates, A. Lavie, et al. 2000. Evaluation of a Practical Interlingua for Task-Oriented Dialogue. In *Workshop on Applied Interlinguas: Practical Applications of Interlingual Approaches to NLP*, Seattle.

Levin, L., D. Gates, A. Lavie, and A. Waibel. 1998. An Interlingua Based on Domain Actions for Machine Translation of Task-Oriented Dialogues. In *Proceedings of ICSLP-98*, Vol. 4, pp. 1155-1158, Sydney, Australia.

Munk, M. 1999. Shallow Statistical Parsing for Machine Translation. Diploma Thesis, Karlsruhe University.

Stevenson, M. and R. Gaizauskas. Experiments on Sentence Boundary Detection. 2000. In *Proceedings of ANLP and NAACL-2000*, Seattle.

Tomita, M. and E. H. Nyberg. 1988. Generation Kit and Transformation Kit, Version 3.2: User's Manual. *Technical Report CMU-CMT-88-MEMO*, Carnegie Mellon University, Pittsburgh, PA.

Woszczyna, M., M. Broadhead, D. Gates, et al. 1998. A Modular Approach to Spoken Language Translation for Large Domains. In *Proceedings of AMTA-98*, Langhorne, PA.