Analysis for Speech Translation Using Grammar-Based Parsing and Automatic Classification

Chad Langley
Language Technologies Institute
Carnegie Mellon University
clangley@cs.cmu.edu

Abstract

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

1 Introduction

For machine translation systems that support many languages, interlingua-based approaches can be very useful. An interlingua defines a language independent representation of the content of utterances. 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. The system then connects a source language analyzer with a target language generator to perform translation.

The analyzer is clearly a critical component in interlingua-based translation systems. For human-to-human speech-to-speech translation systems, the analyzer must be robust to speech recognition errors, spontaneous speech, and ungrammatical inputs (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 languages is important since porting translation systems to new domains or expanding existing coverage can be very time-consuming.

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

I describe a prototype analyzer that combines phrase-level grammar-based parsing and machine learning techniques to take advantage of the benefits of each. The analyzer uses a robust parser and phrase-level semantic grammars to extract low-level arguments from an utterance. Automatic classifiers then assign high-level domain actions to semantic segments in the utterance.

2 MT System Overview

The analyzer that I describe is used for English and German in several multilingual speech-to-speech translation systems, including the NESPOLE! system (Lavie et al., 2002). The goal of NESPOLE! is to provide translation for common users in real-world e-commerce applications such as travel and tourism.

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

3 The Interlingua

The interlingua used 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 dialog. 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 encode specific information from the utterance using a feature-value format. Argument values can be atomic or complex. The IF specification defines all possible components and describes how they can be validly combined. Several examples of utterances with corresponding IF representations 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

The hybrid analysis approach combines grammar-based parsing and machine learning techniques to transform spoken utterances into the interlingua representation. Since the speaker tag is given, the analyzer must 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, automatic classifiers are used to identify the DA.

4.1 Argument Parsing

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

4.1.1 The Parser

SOUP is a stochastic, chart-based, top-down parser 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 critical feature for phrase-level 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 separately 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 version of my analyzer, word skipping is only allowed between parse trees. Also, only the best-ranked argument parse is used.

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 rules correspond to top-level arguments in the interlingua. The pseudo-argument grammar contains rules for parsing common phrases. For example, *all booked up*, *full*, and *sold out* might be grouped into a class of phrases that indicate unavailability. The cross-domain grammar contains rules for parsing complete DAs that are domain-independent. For example, this grammar contains rules for greetings (*Hello*, *Good bye*, *Nice to meet you*, etc.). Finally, the shared grammar contains rules that can be used by all other subgrammars.

4.2 Segmentation

The second stage of processing in my hybrid analysis approach is segmentation of the input into SDUs. In the IF representation, DAs are assigned at the level of SDUs. However, since humans do not generally speak at this level, input utterances must be split into SDUs before assigning DAs.

Figure 1 shows an example of an utterance with 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.

Input to the analyzer is produced by an automatic speech recognizer. Thus, the input contains no punctuation or case information and may contain speech recognition errors. Since the word information around a potential boundary may be unreliable, the segmentation model can also use information derived from the argument parse.

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, and the problem of segmenting an utterance can be divided into subproblems of segmenting the non-cross-domain parts of the utterance. Additionally, the argument parse reduces the number of potential boundary positions because no boundaries are allowed within parse trees. The segmentation model can also look at parse tree labels in the argument parse.

The prototype analyzer drops words skipped between parse trees, leaving only a sequence of trees. The segmenter first examines the grammar label for the roots of the parse trees on each side of a potential SDU boundary position. 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) is used to estimate the likelihood of an SDU boundary. A boundary is inserted when the likelihood exceeds a threshold.

The statistical model is based only on the root labels of the parse trees immediately preceding and following the potential SDU boundary. 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. $C([A_1 \bullet])$ and $C([\bullet A_2])$ are counts of the number of times an SDU boundary followed A_1 or preceded A_2 respectively. $C([A_1])$ and $C([A_2])$ are the total number of times arguments A_1 and A_2 occurred.

4.3 DA Classification

The third stage of analysis in my approach is the identification of the DA for each SDU using automatic classification techniques. Following 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. There are a variety of ways to define the task of identifying the DA. For example, the complete DA could be identified by a single classifier, or the speech act and concept sequence could be classified separately. This would reduce the complexity of each subtask and allow for the application of specialized techniques to each subtask. Likewise, a single classifier could identify the complete concept sequence, or a set of classifiers could be used to indicate the presence of individual concepts in the DA. Independent of the task definition, it would also be possible to apply a variety of classification approaches (memory-based learning, neural networks, language models, etc.). Classification approaches may vary in their suitability to the DA classification task.

In the prototype analyzer, two classifiers are used. The first identifies the speech act, and the second identifies the concept sequence. Both classifiers are implemented using TiMBL (Daelemans *et al*., 2000), which uses memory-based learning (k-NN). Speech act classification is performed first. The speech act classifier takes as input a set of binary features that indicate whether a particular argument label or pseudo-argument label is present in the argument parse for the SDU. No other features are 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 extra feature: the classified speech act.

The analyzer also uses the IF specification to aid classification and guarantee that a valid IF is produced. The speech act and concept sequence classifiers each provide a ranked list of possible classifications. When the best classifications produce an illegal DA, the analyzer attempts to find the next best legal DA. Each of the alternative concept sequences (in ranked order) is combined with each of the alternative speech acts (in ranked order). For each possible 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 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 also fails to produce a legal DA that licenses all of the arguments, the analyzer returns the best-ranked DA that licenses the most arguments. 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 as many arguments as possible than to keep the best DA and lose the detailed information represented by the arguments.

5 Experiments

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

5.1 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 using the NESPOLE! system. The IF specification defined 62 speech acts, 103 concepts, and 147 top-level arguments. The input was a set of English utterances which were paraphrased back into English via the IF (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 `perfect` grade means that the translation was fluent and accurately captured the meaning of the original utterance. An `ok` grade means that the translation was not fluent but nevertheless intelligibly conveyed the meaning of the original utterance. 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, performance was worse with recognition errors.

	Classifier Accuracy (Transcribed Text)
Speech Act	65%
Concept Sequence	54%
Domain Action	43%

Table 3. DA classifier performance

Table 3 shows the performance of the DA classifiers on the same test set. Transcribed utterances were used as input, and the utterances were segmented into SDUs before analysis. In this experiment, the test set contained only 293 SDUs. For the remaining SDUs, it was not possible to assign a valid IF based on the current 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* classification accuracy from Table 3 with the *Transcribed* grades from Tables 1 and 2. Although the DA classifiers produced a perfect DA only 43% of the time, 58% (English) and 55% (Italian) of the translations were graded as acceptable.

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

Table 4. DA elements changed by IF specification

111 (38%) of the 293 SDUs in the test set were assigned by the cross-domain grammar. The remaining 182 SDUs (62%) required DA classification. Table 4 shows how many DAs, speech acts, and concept sequences for these SDUs were changed as a result of using IF 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.

5.2 Ablation Experiment

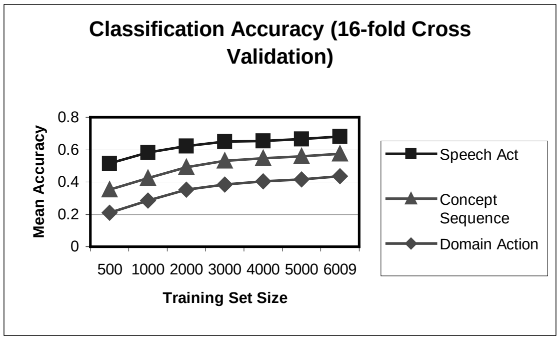

Figure 2. DA classifier accuracy with various amounts of data

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 shown in Figure 2 represents the average accuracy achieved using a 16-fold cross validation setup. This experiment used an earlier version of the analyzer and an IF specification for a different domain.

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 remaining 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 domain action classifiers. A data set this size could be constructed in a few weeks.

6 Related Work

Lavie et al. (1997) developed a method for identifying SDU boundaries in a speech-to-speech translation system. Acoustic information about silences and noises was used. A statistical model that used three word-based bigram frequencies computed from a four-word window was used to estimate the likelihood of an SDU boundary between each pair of words. Lexical cue phrases were used to boost the likelihood estimate.

Identifying SDU boundaries is similar to sentence boundary detection. Stevenson and Gaizauskas (2000) point out that text produced by a speech recognizer differs in important ways from standard text composed by humans. Unlike standard text, speech recognizer output typically contains no punctuation or case information. Furthermore, spoken language often contains phrases and sentence fragments. Finally, speech recognizer output may contain errors. Stevenson and Gaizauskas (2000) use TiMBL (Daelemans et al., 2000) to identify sentence boundaries in automatic 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. Then a neural network identified the concept sequence for each speech act. 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 DA with the highest likelihood is assigned. Arguments are identified using recursive transition networks. IF specification constraints are used to find the most likely valid IF.

7 Future Work

The experimental results indicate the promise of the analysis approach I have described. The level of performance reported here was achieved using a very simple segmentation model and very simple DA classifiers with extremely limited feature sets. I expect the performance of the analyzer will substantially improve through a more informed design of the segmentation model and DA classifiers. I plan to examine various design options, including richer feature sets, alternative classification methods, and using several argument parses rather than just the best-ranked parse.

The primary motivation for developing this approach is to provide improved robustness and portability to new domains and languages. I expect that moving from a purely grammar-based parsing approach to this hybrid approach will help attain this goal by reducing grammar development effort and simplifying annotation requirements. I am planning experiments to evaluate robustness and portability when the coverage of the NESPOLE! translation system is expanded to the medical domain later this year.

8 Acknowledgements

The research work reported here was supported by the National Science Foundation under Grant number 9982227. Special thanks to Alon Lavie, Lori Levin and Alex Waibel for their support on this work and to the grammar writers Dorcas Wallace, Kay Peterson and Donna Gates.

References

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 ASRU 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*. Available from http://ilk.kub.nl/~ilk/papers/ilk0001.ps.gz.

Gavaldà, M. 2000. SOUP: A Parser for Real-World Spontaneous Speech. In *Proceedings of 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., 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, F. Pianesi, D. Wallace, T. Watanabe, and M. Woszczyna. 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.

Woszczyna, M., M. Broadhead, D. Gates, M. Gavaldà, A. Lavie, L. Levin, and A. Waibel. 1998. A Modular Approach to Spoken Language Translation for Large Domains. In *Proceedings of AMTA-98*, Langhorne, PA.