PARSING DOMAIN ACTIONS WITH PHRASE-LEVEL GRAMMARS AND MEMORY-BASED LEARNERS

Chad Langley and **Alon Lavie**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{clangley|alavie}@cs.cmu.edu

Abstract

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

1 Introduction

Interlingua-based approaches to machine translation (MT) are very attractive for systems that support multiple languages. An interlingua defines a language independent representation of the content of utterances. For each source language, an analyzer must convert input utterances into the interlingua representation. Likewise, for each target language, a generator must convert the interlingua into target language output. Translation is performed by connecting a source language analyzer with a target language generator.

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 often provides very accurate analyses, it is generally not feasible to develop a grammar that completely covers a domain, and this problem is further exacerbated with spoken input, where disfluent input and deviations from the grammar are very common. Furthermore, a great deal of effort by human experts is generally required to develop a wide-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 than grammars on common in-domain input, and may require a large amount of training data in order to achieve adequate levels of performance.

In this paper, we describe an analyzer that combines phrase-level grammar-based parsing and machine learning techniques in a way that leverages from 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 are then used to segment the utterance and to assign high-level domain actions to each semantic segment.

2 MT System Overview

The analyzer that we describe is used for English and German in NESPOLE! (Lavie et al., 2002), a multilingual speech-to-speech machine translation system. The goal of NESPOLE! is to provide speech-translation for common users engaged in real-world e-commerce applications such as travel and tourism. NESPOLE! translates via an interlingua-based approach in four basic steps: (1) an automatic speech recognizer processes the spoken input; (2) the best-ranked text hypothesis from speech recognition is processed by the analyzer, producing an interlingua representation; (3) target language text is generated from the interlingua; and (4) the text is synthesized into speech.

3 The Interlingua

The interlingua we use 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 tag>:<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 Interchange Format representation. The speaker tag is configured by the system, and the analyzer must identify the domain action 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 using memory-based learning (k-nearest neighbor) techniques. Finally, additional memory-based classifiers are used to identify the domain action (speech-act and sequence of concepts).

4.1 Argument Parsing

The first step in our analysis approach is to parse an utterance for arguments. 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. SOUP provides several features that are useful for phrase-level argument parsing. One important feature provided by SOUP is word skipping. The amount of skipping allowed is configurable, and a list of words which cannot be skipped may 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 version of the analyzer described here, word skipping is only allowed between parse trees, and 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 Interchange Format. Top-level argument grammar rules correspond to top-level arguments in the IF. The pseudo-argument grammar contains rules for parsing common phrases that are not covered by IF arguments. 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 low-level rules that can be used by all other subgrammars.

4.2 Segmentation

The second stage of processing in our hybrid analysis approach is segmentation of the input into SDUs. In the IF representation, DAs are assigned at the level of SDUs. Speech turns, however, often consist of several SDUs, and thus must be segmented before assigning DAs. Figure 1 shows an example of an utterance with four arguments segmented into two SDUs.

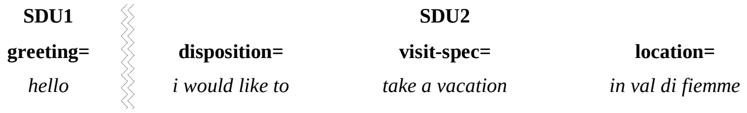

Figure 1: Segmentation of an utterance into SDUs

Since the input to the analyzer is text produced by an automatic speech recognizer, neither punctuation nor case information are explicitly represented, and speech recognition errors may be present. In addition to the word information surrounding a potential SDU boundary, which may be unreliable, the segmenter also uses 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 parts of the utterance not covered by a cross-domain tree. Additionally, SDU boundaries cannot occur within parse trees. Thus, potential SDU boundary positions can be hypothesized only between parse trees and/or unparsed words. The segmenter also uses the root labels of argument parses.

The segmenter in the version of the analyzer described here is implemented using TiMBL (Daelemans et al., 2002), a memory-based (k-Nearest-Neighbor) learning program. 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, the TiMBL segmentation classifier uses ten features based on the words and arguments surrounding the potential boundary to determine if an SDU boundary is present.

The features used by the TiMBL classifier include the word and argument parse tree label immediately preceding and immediately following the potential boundary (w_{-1}, w_1, A_{-1}, and A_1). When an unparsed word occurs on either side of a potential boundary, a token indicating an unparsed word is used in place of a true argument label. In addition, the probabilities that a boundary follows the preceding word and argument label ($\mathrm{P}(w_{-1}\bullet)$ and $\mathrm{P}(A_{-1}\bullet)$) and precedes the following word and argument label ($\mathrm{P}(\bullet w_1)$ and $\mathrm{P}(\bullet A_1)$) are used as input features. These probabilities are computed based on counts from the training data (i.e., $\mathrm{P}(w_{-1}\bullet) = \mathrm{C}(w_{-1}\bullet)/\mathrm{C}(w_{-1})$). The final two features are the number of words since the last boundary and the number of argument parse trees since the last boundary.

The training data for the segmentation classifier are extracted from utterances that have been annotated with SDU boundaries and parsed using the phrase-level argument parser. A training example is created for each potential boundary position in the parsed data. Positive segmentation examples occur between SDUs, as marked in the data. Negative segmentation examples occur within an SDU.

For example, in the utterance shown in Figure 1, the potential boundary position between “*hello*” and “*i*” would be a positive segmentation example. The potential boundary position between “*to*” and “*take*” would be a negative segmentation example.

4.3 Domain Action Classification

The third stage of analysis 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.

The version of the analyzer described here uses two classifiers to determine the DA for non-cross-domain SDUs. The first classifier identifies the speech act, and the second identifies the complete concept sequence. Both classifiers are implemented using TiMBL (Daelemans et al., 2002). Speech act classification is performed first. The speech act classifier takes as input a set of binary features that indicate whether each of the various argument labels and pseudo-argument labels is present in the argument parse forest of the SDU. No other features are used. Concept sequence classification is performed after speech act classification. The concept sequence classifier uses the same features as the speech act classifier with one extra feature: the 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 combination of the top-ranked speech-act and the top-ranked concept sequence results in an illegal DA, the analyzer attempts to find an alternative 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 since illegal arguments may cause a generation failure. The rationale behind this approach is that it is generally preferable to select a lower-ranking DA and retain as many arguments as possible, rather than selecting the top-ranked DA and losing some detailed information represented by the arguments.

5 Evaluation

We present the results from recent experiments to assess the performance of the segmenter and DA classifiers individually and of end-to-end translation using the analyzer. We conducted classification experiments using data from two domains: travel/tourism (vacation planning) and medical (doctor/patient diagnosis). The data for the classification experiments consisted primarily of domain specific dialogues that were collected monolingually. The data were manually transcribed, segmented into semantic dialogue units, and tagged with Interchange Format representations.

We also evaluated end-to-end translation performance on the travel/tourism domain. The test data for the end-to-end translation experiment consisted of 2 previously unseen dialogues. Our hybrid analyzer was used to segment the best hypothesis from automatic speech recognition into SDUs and label each SDU with an IF representation. Target language text was then generated from the IF representations. Thus, the results of the end-to-end evaluation reflect the combined performance of the recognizer, analyzer, and generator.

5.1 Segmentation

	English Travel	German Travel	English Medical	German Medical
Accuracy	0.9480	0.9513	0.9646	0.9393
Precision+	0.9571	0.9640	0.9341	0.9223
Recall+	0.9301	0.9358	0.9743	0.9559
Training Examples	35417	45945	41889	7674

Table 1: Segmentation classifier performance

Table 1 shows the performance of the TiMBL segmentation classifier for English and German. The results reported in Table 1 were computed over the training data using the leave-one-out testing method provided by the TiMBL software. In the leave-one-out method, each example is held out from the training set and classified using all of the remaining examples for training. The segmentation classifier used the IB1 (k-NN) algorithm with 5 neighbors, unweighted voting, and Gain Ratio feature weighting (Daelemans et al., 2002). The *Accuracy* row shows the overall classification accuracy over all possible boundary positions. The *Precision+* and *Recall+* rows are computed only over positions where a boundary is present.

5.2 Domain Action Classification

	English Travel	German Travel	English Medical	German Medical
Accuracy	0.7001	0.6755	0.7814	0.6892
Training Examples	8289	8719	3659	2294

Table 2: Speech act classifier performance

	English Travel	German Travel	English Medical	German Medical
Accuracy	0.6984	0.6719	0.6464	0.6997
Training Examples	8289	8719	3659	2294

Table 3: Concept sequence classifier performance

Tables 2 and 3 show the performance of the TiMBL speech act and concept sequence classifiers for English and German. As in the segmentation experiment, the results were computed using the leave-one-out method. The classifiers used the IGTREE algorithm (Daelemans et al., 2002).

5.3 End-to-End Translation

English WAR	German WAR
56.4%	51.0%

Table 4: Speech Recognition Word Accuracy Rates

	English Output	Italian Output
SR Hypotheses	66.7%	--
Translation from SR Hypotheses	50.4%	50.2%

Table 5: Acceptable end-to-end translation for English travel input

	German Output	Italian Output
SR Hypotheses	61.6%	--
Translation from SR Hypotheses	53.4%	51.7%

Table 6: Acceptable end-to-end translation from German travel input

Table 4 shows the word accuracy rates on the test data for the automatic speech recognizers for English and German. Tables 5 and 6 show end-to-end translation results from the most recent evaluation of the NESPOLE! system for English and German input. The data used to train the segmentation and DA classifiers were the same as in the classification experiments. The English test set contained 110 utterances consisting of 232 SDUs from 2 unseen dialogues. The German test set contained 246 utterances consisting of 356 SDUs from 2 unseen dialogues. Translations were compared to human transcriptions and graded by 3 human graders using a 4-point scale with grades of *perfect*, *ok*, *bad*, and *very bad*. A grade of *perfect* or *ok* is considered *Acceptable*. A grade of *bad* or *very bad* is considered *Unacceptable*. For each source language, graders evaluated both "translations" of the input back into the source language and translations into Italian. The results shown in the table were produced using a majority vote among the 3 graders for each SDU. The translation for an SDU was considered *Acceptable* if at least 2 of the graders graded it as such.

The row labeled *SR Hypotheses* shows the grades when the speech recognizer output is compared directly to human transcripts (i.e. when the SR output is treated as a translation back into the source language). As these grades show, recognition errors can be a major source of unacceptable translations. These SR grades provide a rough bound on the translation performance that can be expected when translating input from the speech recognizer since meaning lost due to recognition errors cannot be recovered by the analyzer. The row labeled *Translation from SR Hypotheses* shows the performance when the speech recognizer produces the input utterances. These grades reflect the combined performance of the speech recognizer, all of the analyzer components, and the generator.

6 Related Work

Lavie et al. (1997) developed a method for identifying SDU boundaries in a speech-to-speech translation system. The method combines acoustic information about silences and noises with a statistical model that uses three word-based bigram frequencies computed from a four-word window, in order to estimate the likelihood of an SDU boundary between each pair of words. Lexical cue phrases were then 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 an 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 Discussion

The experimental results indicate the promise of our hybrid analysis approach. The memory-based segmentation classifier provides reasonable performance on both domains for both English and German. The DA classification performance reported here was achieved using classifiers with a very simple feature set. We expect that the performance of the DA classifiers can be improved by including additional features. For example, the classifiers described here used only the label of the argument parse trees. Since pseudo-argument grammar trees may contain real IF arguments as subtrees, information may be gained by extracting any real arguments from the pseudo-argument trees. Additionally, word information or context information, such as the previous DA from each speaker may provide useful information for classification. Thus, we plan to examine the effects of richer feature sets on DA classification.

We also plan to examine alternative definitions of the DA classification problem. For the work reported here, we chose to identify the speech act with one classifier and the complete concept sequence with another. However, it would also be possible to identify the complete DA with a single classifier or to use separate classifiers for each individual concept. Independent of the DA classification method, it would also be possible to apply a variety of alternative classification techniques (i.e. neural networks, language models, etc.). We will perform a comparison of the memory-based approach described here with other classification approaches.

The primary motivation for developing this approach was to provide improved robustness and portability to new domains and languages. In order to build a translator for a new domain, the speech recognizer, interlingua, analyzer, and generator must be updated to cover the new domain. The effort required to port the analyzer is independent of the effort required to port the other components, assuming that the interlingua retains the same format and only adds new domain specific components. We 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. In our approach, grammars must only be written for new domain specific arguments, and utterances must be tagged with domain actions in order to train new classifiers.

We are currently working on evaluating the portability of our hybrid approach by expanding coverage into the medical doctor/patient diagnosis domain. Starting with the existing travel domain grammars, approximately 90 person-hours were spent expanding the English grammars to cover the medical domain. Approximately 50 person-hours were spent expanding the existing German grammars. In addition approximately 15 person-hours were spent segmenting and tagging the data with domain actions. As the results from the classification experiments show, the performance of the classifiers for the medical domain was similar to that of the travel domain. We are in the process of developing full grammars for the medical domain in order to assess the development effort required for writing full grammars and to compare the performance of the full grammar approach with our hybrid approach.

8 Acknowledgements

The research work reported here was supported by the National Science Foundation under Grant number 9982227. Special thanks to 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. 2002. TiMBL: Tilburg Memory Based Learner, version 4.3, Reference Guide. ILK Technical Report 02-10. Available from http://ilk.kub.nl/downloads/pub/papers/ilk0210.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.

Langley, C., A. Lavie, L. Levin, D. Wallace, D. Gates, and K. Peterson. 2002. Spoken Language Parsing Using Phrase-Level Grammars and Trainable Classifiers. In Workshop on Algorithms for Speech-to-Speech Machine Translation at ACL-02, Philadelphia, PA.

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.