Domain Specific Speech Acts for Spoken Language Translation

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

Abstract

We describe a coding scheme for machine translation of spoken task-oriented dialogue. The coding scheme covers two levels of speaker intention – domain independent speech acts and domain dependent domain actions. Our database contains over 14,000 tagged sentences in English, Italian, and German. We argue that domain actions, and not speech acts, are the relevant discourse unit for improving translation quality. We also show that, although domain actions are domain specific, the approach scales up to large domains without an explosion of domain actions and can be coded with high inter-coder reliability across research sites. Furthermore, although the number of domain actions is on the order of ten times the number of speech acts, sparseness is not a problem for the training of classifiers for identifying the domain action. We describe our work on developing high accuracy speech act and domain action classifiers, which is the core of the source language analysis module of our NESPOLE machine translation system.

1 Introduction

The NESPOLE and C-STAR machine translation projects use an interlingua representation based on speaker intention rather than literal meaning. The speaker's intention is represented as a domain independent speech act followed by domain dependent concepts. We use the term *domain action* to refer to the combination of a speech act with domain specific concepts. Examples of domain actions and speech acts are shown in Figure 1.

```
c:give-information+party
“I will be traveling with my husband and
our two children ages two and eleven”

c:request-information+existence+facility
“Do they have parking available?"
“Is there someplace to go ice skating?"

c:give-information+view+information-
object
“I see the bus icon”
```

Figure 1: Examples of Speech Acts and Domain Actions.

Domain actions are constructed compositionally from an inventory of speech acts and an inventory of concepts. The allowable combinations of speech acts and concepts are formalized in a human- and machine-readable specification document. The specification document is supported by a database of over 14,000 tagged sentences in English, German, and Italian.

The discourse community has long recognized the potential for improving NLP systems by identifying speaker intention. It has been hypothesized that predicting speaker intention of the next utterance would improve speech recognition (Reithinger et al., Stolcke et al.), or reduce ambiguity for machine translation (Qu et al., 1996, Qu et al., 1997). Identifying speaker intention is also critical for sentence generation.

We argue in this paper that the explicit representation of speaker intention using domain actions can serve as the basis for an effective language-independent representation of meaning for speech-to-speech translation and that the relevant units of speaker intention are the domain specific domain action as well as the domain independent speech act. After a brief description of our database, we present linguistic motivation for domain actions. We go on to show that although domain actions are domain specific, there is not an explosion or exponential growth of domain actions when we scale up to a larger domain or port to a new domain. Finally we will show that, although the number of domain actions is on the

order of ten times the number of speech acts, data sparseness is not a problem in training a domain action classifier. We present extensive work on developing a high-accuracy classifier for domain actions using a variety of classification approaches and conclusions on the adequacy of these approaches to the task of domain action classification.

2 Data Collection Scenario and Database

Our study is based on data that was collected for the NESPOLE and C-STAR speech-to-speech translation projects. Three domains are included. The NESPOLE travel domain covers inquiries about vacation packages. The C-STAR travel domain consists largely of reservation and payment dialogues and overlaps only about 50% in vocabulary with the NESPOLE travel domain. The medical assistance domain includes dialogues about chest pain and flu-like symptoms.

There were two data collection protocols for the NESPOLE travel domain – monolingual and bilingual. In the monolingual protocol, an English speaker in the United States had a conversation with an Italian travel agent speaking (non-native) English in Italy. Monolingual data was also collected for German, French and Italian. Bilingual data was collected during user studies with, for example, an English speaker in the United States talking to an Italian-speaking travel agent in Italy, with the NESPOLE system providing the translation between the two parties. The C-STAR data consists of only monolingual role-playing dialogues with both speakers at the same site. The medical dialogues are monolingual with doctors playing the parts of both doctor and patient.

The dialogues were transcribed and multi-sentence utterances were broken down into multiple Semantic Dialogue Units (SDUs) that each correspond to one domain action. Some SDUs have been translated into other NESPOLE or C-STAR languages. Over 14,000 SDUs have been tagged with interlingua representations including domain actions as well as argument-value pairs. Table 1 summarizes the number of tagged SDUs in complete dialogues in the interlingua database. There are some additional tagged dialogue fragments that are not counted. Figure 2 shows an excerpt from the database.

English	NESPOLE Travel	4691
English	C-STAR Travel	2025
German	NESPOLE Travel	1538
Italian	NESPOLE Travel	2248
English	Medical Assistance	2001
German	Medical Assistance	1152
Italian	Medical Assistance	935

Table 1: Tagged SDUs in the Interlingua Database.

```
e709wa.19.0  comments: DATA from
e709_1_0018_ITAGOR_00

e709wa.19.1  olang ITA  lang ITA Prv CMU
“hai in mente una localita specifica?"
e709wa.19.1  olang ITA  lang GER  Prv CMU
“haben Sie einen bestimmten Ort im Sinn?"
e709wa.19.1  olang ITA  lang FRE  Prv
CLIPS “"
e709wa.19.1  olang ITA  lang ENG  Prv CMU
“do you have a specific place in mind"
e709wa.19.1                    IF  Prv CMU
a:request-information+disposition+object
(object-spec=(place, modifier=specific,
identifiability=no), disposi-
tion=(intention, who=you))
e709wa.19.1  comments: Tagged by dmg
```

Figure 2: Excerpt from the Interlingua Database.

3 Linguistic Argument for Domain Actions

Proponents of Construction Grammar (Fillmore et. al. 1988, Goldberg 1995) have argued that human languages consist of constructional units that include a syntactic structure along with its associated semantics and pragmatics. Some constructions follow the typical syntactic rules of the language but have a semantic or pragmatic focus that is not compositionally predictable from the parts. Other constructions do not even follow the typical syntax of the language (e.g., *Why not go?* with no tensed verb).

Our work with multilingual machine translation of spoken language shows that fixed expressions cannot be translated literally. For example,

Why not go to the meeting? can be translated into Japanese as *Kaigi ni itte mitara doo?* (meeting to going see/try-if how), which differs from the English in several ways. It does not have a word corresponding to *not*; it has a word that means *see/try* that does not appear in the English sentence; and so on. In order to produce an acceptable translation, we must find a common ground between the English fixed expression *Why not V-inf?* and the Japanese fixed expression *-te mittara doo?*. The common ground is the

speaker's intention (in this case, to make a suggestion) rather than the syntax or literal meaning.

Speaker intention is partially captured with a direct or indirect speech act. However, whereas speech acts are generally domain independent, task-oriented language abounds with fixed expressions that have domain specific functions. For example, the phrases *We have…* or *There are…* in the hotel reservation domain express availability of rooms in addition to their more literal meanings of possession and existence. In the past six years, we have been successful in using domain specific domain actions as the basis for translation of limited-domain task-oriented spoken language (Levin et al., 1998, Levin et al. 2002; Langley and Lavie, 2003)

4 Scalability and Portability of Domain Actions

Domain actions, like speech acts, convey speaker intention. However, domain actions also represent components of meaning and are therefore more numerous than domain independent speech acts. 1168 unique domain actions are used in our NESPOLE database, in contrast to only 72 speech acts. We show in this section that domain actions yield good coverage of task-oriented domains, that domain actions can be coded effectively by humans, and that scaling up to larger domains or porting to new domains is feasible without an explosion of domain actions.

Coverage of Task-Oriented Domains: Our NESPOLE domain action database contains dialogues from two task-oriented domains: medical assistance and travel. Table 2 shows the number of speech acts and concepts that are used in the travel and medical domains. The 1168 unique domain actions that appear in our database are composed of the 72 speech acts and 125 concepts.

	Travel	**Medical**	**Combined**
DAs	880	459	1168
SAs	67	44	72
Concepts	91	74	125

Table 2: DA component counts in NESPOLE data.

Our domain action based interlingua has quite high coverage of the travel and medical dialogues we have collected. To measure how well the interlingua covers a domain, we define the *no-tag* rate as the percent of sentences that are not covered by the interlingua, according to a human expert. The no-tag rate for the English NESPOLE travel dialogues is 4.3% for dialogues that have been used for system development.

We have also estimated the domain action no-tag rate for unseen data using the NESPOLE travel database (English, German, and Italian combined). We randomly selected 100 SDUs as seen data and extracted their domain actions. We then randomly selected 100 additional SDUs from the remaining data and estimated the no-tag rate by counting the number of SDUs not covered by the domain actions in the seen data. We then added the unseen data to the seen data set and randomly selected 100 new SDUs. We repeated this process until the entire database had been seen, and we repeated the entire sampling process 10 times. Although the number of domain actions increases steadily with the database size (Figure 4), the no-tag rate for unseen data stabilizes at less than 10%.

We also randomly selected half of the SDUs (4200) from the database as seen data and extracted the domain actions. Holding the seen data set fixed, we then estimated the no-tag rates in increasing amounts of unseen data from the remaining half of the database. We repeated this process 10 times. With a fixed amount of seen data, the no-tag rate remains stable for increasing amounts of unseen data. We observed similar no-tag rate results for the medical assistance domain and for the combination of travel and medical domains.

It is also important to note that although there is a large set of uncommon domain actions, the top 105 domain actions cover 80% of the sentences in the travel domain database. Thus domain actions are practical for covering task-oriented domains.

Intercoder Agreement: Intercoder agreement is another indicator of manageability of the domain action based interlingua. We calculate intercoder agreement as percent agreement. Three interlingua experts at one NESPOLE site achieved 94% agreement (average pairwise agreement) on speech acts and 88% agreement on domain actions. Across sites, expert agreement on speech acts is still quite high (89%), although agreement on domain actions is lower (62%). Since many domain actions are similar in meaning, some disagreement can be tolerated without affecting translation quality.

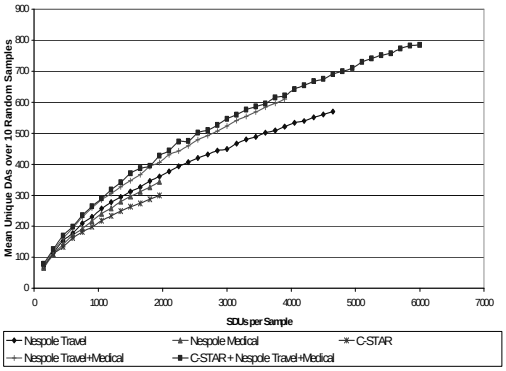

Figure 3: DAs to cover data (English).

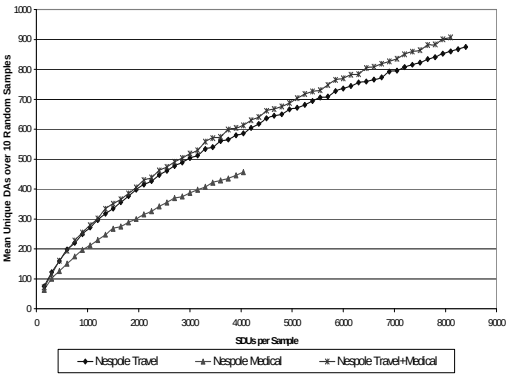

Figure 4: DAs to cover data (All languages).

Scalability and Portability: The graphs in Figure 3 and Figure 4 illustrate growth in the number of domain actions as the database size increases and as new domains are added. The x-axis represents the sample size randomly selected from the database. The y-axis shows the number of unique domain actions (types) averaged over 10 samples of each size. Figure 3 shows the growth in domain actions for three English databases (NESPOLE travel, C-STAR travel, and medical assistance) as well as the growth in domain actions for a database consisting of equal amounts of data from each domain. Figure 4 shows the growth in domain actions for combined English, German, and Italian data in the NESPOLE travel and medical domains.

Figure 3 and Figure 4 show that the number of domain actions increases steadily as the database grows. However, closer examination reveals that scalability to larger domains and portability to new domains are in fact feasible. The curves representing combined domains (travel plus medical in Figure 4 and NESPOLE travel, C-STAR travel, and medical in Figure 3) show only a small increase in the number of domain actions when two domains are combined. In fact, there is a large overlap between domains. In Table 3 the Overlap columns show the number of DA types and tokens that are shared between the travel and medical domains. We can see around 70% of DA tokens are covered by DA types that occur in both domains.

	DA Types	Type Overlap	DA Tokens	Token Overlap
NESPOLE Travel	880	171	8477	6004 (70.8%)
NESPOLE Medical	459	171	4088	2743 (67.1%)

Table 3: DA Overlap (All languages).

5 A Hybrid Analysis Approach for Parsing Domain Actions

Langley et al. (2002; Langley and Lavie, 2003) describe the hybrid analysis approach that is used in the NESPOLE! system (Lavie et al., 2002). The hybrid analysis approach combines grammar-based phrasal parsing and machine learning techniques to transform utterances into our interlingua representation. Our analyzer operates in three stages to identify the domain action and arguments.

First, an input utterance is parsed into a sequence of arguments using phrase-level semantic grammars and the SOUP parser (Gavaldà, 2000). 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 interlingua. The pseudo-argument grammar contains rules for parsing common phrases that are not covered by interlingua 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.

After argument parsing, the utterance is segmented into SDUs using memory-based learning (k-nearest neighbor) techniques. Spoken utterances often consist of several SDUs. Since DAs are assigned at the SDU level, it is necessary to segment utterances before assigning DAs.

The final stage in the hybrid analysis approach is domain action classification.

6 Domain Action Classification

Identifying the domain action is a critical step in the analysis process for our interlingua-based translation systems. One possible approach would be to manually develop grammars designed to parse input utterances all the way to the domain action level. However, while grammar-based parsing may provide very accurate analyses, it is generally not feasible to develop a grammar that completely covers a domain. This problem is exacerbated with spoken input, where disfluencies 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.

An alternative to writing full domain action grammars is to train classifiers to identify the DA. Machine learning approaches allow the analyzer to 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, especially on common in-domain input, and may require a large amount of training data in order to achieve adequate levels of performance. In the hybrid analyzer described above, classifiers are used to identify the DA for domain specific portions of utterances that are not covered by the cross-domain grammar.

We tested classifiers trained to classify complete DAs. We also split the DA classification task into two subtasks: speech act classification and concept sequence classification. This simplifies the task of each classifier, allows for the use of different approaches and/or feature sets for each task, and reduces data sparseness. Our hybrid analyzer uses the output of each classifier along with the interlingua specification to identify the DA (Langley et al., 2002; Langley and Lavie, 2003).

7 Experimental Setup

We conducted experiments to assess the performance of several machine-learning approaches on the DA classification tasks. We evaluated all of the classifiers on English and German input in the NESPOLE travel domain.

7.1 Corpus

The corpus used in all of the experiments was the NESPOLE! travel and tourism database. Since our goal was to evaluate the SA and concept sequence classifiers and not segmentation, we created training examples for each SDU in the database rather than for each utterance. Table 4 contains statistics regarding the contents of the corpus for our classification tasks. Table 5 shows the frequency of the most common domain action, speech act, and concept sequence in the corpus. These frequencies provide a baseline that would be achieved by a simple classifier that always returned the most common class.

	English	German
SDUs	8289	8719
Domain Actions	972	1001
Speech Acts	70	70
Concept Sequences	615	638
Vocabulary Size	1946	2815

Table 4: Corpus Statistics.

	English	German
DA (*acknowledge*)	19.2%	19.7%
SA (*give-information*)	41.4%	40.7%
Concept Sequence (No concepts)	38.9%	40.3%

Table 5: Most frequent DAs, SAs, and CSs.

All of the results presented in this paper were produced using a 20-fold cross validation setup. The corpus was randomly divided into 20 sets of equal size. Each of the sets was held out as the test set for one fold with the remaining 19 sets used as training data. Within each language, the same random split was used for all of the classification experiments. Because the same split of the data was used for different classifiers, the results of two classifiers on the same test set are directly comparable. Thus, we tested for significance using two-tailed matched pair t-tests.

7.2 Machine Learning Approaches

We evaluated the performance of four different machine-learning approaches on the DA classification tasks: memory-based learning (k-Nearest-Neighbor), decision trees, neural networks, and naïve Bayes n-gram classifiers. We selected these approaches because they vary substantially in the their representations of the training data and their methods for selecting the best class.

Our purpose was not to implement each approach from scratch but to test the approach for our particular task. Thus, we chose to use existing software for each approach "off the shelf." The ease of acquiring and setting up the software influenced our choice. Furthermore, the ease of incorporating the software into our online translation system was also a factor.

Our memory-based classifiers were implemented using TiMBL (Daelemans et al., 2002). We used C4.5 (Quinlan, 1993) for our decision tree classifiers. Our neural network classifiers were implemented using SNNS (Zell et al., 1998). We used Rainbow (McCallum, 1996) for our naïve Bayes n-gram classifiers.

8 Experiments

In our first experiment, we compared the performance of the four machine learning approaches. Each SDU was parsed using the argument and pseudo-argument grammars described above. The feature set for the DA and SA classifiers consisted of binary features indicating the presence or absence of labels from the grammars in the parse forest for the SDU. The feature set included 212 features for English and 259 features for German. The concept sequence classifiers used the same feature set with the addition of the speech act.

In the SA classification experiment, the TiMBL classifier used the IB1 (k-NN) algorithm with 1 neighbor and gain ratio feature weighting. The C4.5 classifier required at least one instance per branch and used node post-pruning. Both the TiMBL and C4.5 classifiers used the binary features described above and produced the single best class as output. The SNNS classifier used a simple feed-forward network with 1 input unit for each binary feature, 1 hidden layer containing 15 units, and 1 output unit for each speech act. The network was trained using backpropagation. The order of presentation of the training examples was randomized in each epoch, and the weights were updated after each training example presentation. In order to simulate the binary features used by the other classifiers as closely as possible, the Rainbow classifier used a simple unigram model whose vocabulary was the set of labels included in the binary feature set. The setup for the DA classification experiment was identical except that the neural network had 50 hidden units.

The setup of the classifiers for the concept sequence classification experiment was very similar. The TiMBL and C4.5 classifiers were set up exactly as in the DA and SA experiments with one extra feature whose value was the speech act. The SNNS concept sequence classifier used a similar network with 50 hidden units. The SA feature was represented as a set of binary input units. The Rainbow classifier was set up exactly as in the DA and SA experiments. The SA feature was not included.

As mentioned above, both experiments used a 20-fold cross-validation setup. In each fold, the TiMBL, C4.5, and Rainbow classifiers were simply trained on 19 subsets of the data and tested on the remaining set. The SNNS classifiers required a more complex setup to determine the number of epochs to train the neural network for each test set. Within each fold, a cross-validation setup was used to determine the number of training epochs. Each of the 19 training subsets for a fold was used as a validation set. The network was trained on the remaining 18 subsets until the accuracy on the validation set did not improve for 50 consecutive epochs. The network was then trained on all 19 training subsets for the average number of epochs from the validation sets. This process was used for all 20-folds in the SA classification experiment. For the DA and concept sequence experiments, this process ran for approximately 1.5 days for each fold. Thus, this process was run for the first two folds, and the average number of epochs from those folds was used for training.

	English	German
TiMBL	49.69%	46.51%
C4.5	48.90%	46.58%
SNNS	49.39%	46.21%
Rainbow	39.74%	38.32%

Table 6: Domain Action classifier accuracy.

	English	German
TiMBL	69.82%	67.57%
C4.5	70.41%	67.90%
SNNS	71.52%	67.61%
Rainbow	51.39%	46.00%

Table 7: Speech Act classifier accuracy.

	English	German
TiMBL	69.59%	67.08%
C4.5	68.47%	66.45%
SNNS	71.35%	68.67%
Rainbow	51.64%	51.50%

Table 8: Concept Sequence classifier accuracy.

Table 6, Table 7, and Table 8 show the average accuracy of each learning approach on the 20-fold cross validation experiments for domain action, speech act, and concept classification respectively. For DA classification, there were no significant differences between the TiMBL, C4.5, and SNNS classifiers for English or German. In the SA experiment, the difference between the TiMBL and C4.5 classifiers for English was not significant. The SNNS classifier was significantly better than both TiMBL and C4.5 (at least p=0.0001). For German SA classification, there were no significant differences between the TiMBL, C4.5, and SNNS classifiers. For concept sequence classification, SNNS was significantly better than TiMBL and C4.5 (at least p=0.0001) for both English and German. For English only, TiMBL was significantly better than C4.5 (p=0.005).

For both languages, the Rainbow classifier performed much worse than the other classifiers. However, the unigram model over arguments did not exploit the strengths of the n-gram classification approach. Thus, we ran another experiment in which the Rainbow classifier was trained on simple word bigrams. No stemming or stop words were used in building the bigram models.

	English	German
Domain Action	48.59%	48.09%
Speech Act	79.00%	77.46%
Concept Sequence	56.87%	57.77%

Table 9: Rainbow accuracy with word bigrams.

Table 9 shows the average accuracy of the Rainbow word bigram classifiers using the same 20-fold cross-validation setup as in the previous experiments. As we expected, using word bigrams rather than parse label unigrams improved the performance of the Rainbow classifiers. For German DA classification, the word bigram classifier was significantly better than all of the previous German DA classifiers (at least p=0.005). Furthermore, the Rainbow word bigram SA classifiers for both languages outperformed all of the SA classifiers that used only the parse labels.

Although the argument parse labels provide an abstraction of the words present in an SDU, the words themselves also clearly provided useful information for classification, at least for the SA task. Thus, we conducted additional experiments to examine whether combining parse and word information could further improve performance.

We chose to incorporate word information into the TiMBL classifiers used in the first experiment. Although the SNNS SA classifier performed significantly better than the TiMBL SA classifier for English, there was no significant difference for SA classification in German. Furthermore, because of the complexity and time required for training with SNNS, we preferred working with TiMBL.

We tested two approaches to adding word information to the TiMBL classifier. In both approaches, the word-based information for each fold was computed only based on the data in the training set. In our first approach, we added binary features for the 250 words that had the highest mutual information with the class. Each feature indicated the presence or absence of the word in the SDU. In this condition, we used the TiMBL classifier with gain ratio feature weighting, 3 neighbors, and unweighted voting. The second approach we tested combined the Rainbow word bigram classifier with the TiMBL classifier. We added one input feature for each possible speech act to the TiMBL classifier. The value of each SA feature was the probability of the speech act computed by the Rainbow word bigram classifier. In this condition, we used the TiMBL classifier with gain ratio feature weighting, 11 neighbors, and inverse linear distance weighted voting.

	English	German
TiMBL + words	78.59%	75.98%
TiMBL + Rainbow	81.25%	78.93%

Table 10: Word+Parse SA classifier accuracy.

Table 10 shows the average accuracy of the SA classifiers that combined parse and word information using the same 20-fold cross-validation setup as the previous experiments. Although adding binary features for individual words improved performance over the classifiers with no word information, it did not allow the combined classifiers to outperform the Rainbow word bigram classifiers. However, for both languages, adding the probabilities computed by the Rainbow bigram model resulted in a SA classifier that outperformed all previous classifiers. The improvement in accuracy was highly significant for both languages.

We conducted a similar experiment for combining parse and word information in the concept sequence classifiers. The first condition was

analogous to the first condition in the combined SA classification experiment. The second condition was slightly different. A concept sequence can be broken down into a set of individual concepts. The set of individual concepts is much smaller than the set of concept sequences (110 for English and 111 for German). Thus, we used a Rainbow word bigram classifier to compute the probability of each individual concept rather than the complete concept sequence. The probabilities for the individual concepts were added to the parse label features for the combined classifier. In both conditions, the performance of the combined classifiers was roughly the same as the classifiers that used only parse labels as features.

	English	German
TiMBL + words	56.48%	54.98%

Table 11: Word+Parse DA classifier accuracy.

Table 11 shows the average accuracy of DA classifiers for English and German using a setup similar to the first approach in the combined SA experiment. In this experiment, we added binary features for the 250 words that the highest mutual information with the class. We used a TiMBL classifier with gain ratio feature weighting and one neighbor. The improvement in accuracy for both languages was highly significant.

	English	German
TiMBL SA + TiMBL CS	49.63%	46.50%
TiMBL+Rainbow SA + TiMBL CS	57.74%	53.93%

Table 12: DA accuracy of SA+CS classifiers.

Finally, Table 12 shows the results from two tests to compare the performance of combining the best output of the SA and concept sequence classifiers with the performance of the complete DA classifiers. In the first test, we combined the output from the TiMBL SA and CS classifiers shown in Table 7 and Table 8. The performance of the combined SA+CS classifiers was almost identical to that of the TiMBL DA classifiers shown in Table 6. In the second test, we combined our best SA classifier (TiMBL+Rainbow, shown in Table 10) with the TiMBL CS classifier. In this case, we had mixed results. The performance of the combined classifiers was better than our best DA classifier for English and worse for German.

9 Discussion

One of our main goals was to determine the feasibility of automatically classifying domain actions. As the data in Table 4 show, DA classification is a challenging problem with approximately 1000 classes. Even when the task is divided into subproblems of identifying the SA and concept sequence, the subtasks remain difficult. The difficulty is compounded by relatively sparse training data with unevenly distributed classes. Although the most common classes in our training corpus had over 1000 training examples, many of the classes had only 1 or 2 examples.

Despite these difficulties, our results indicate that domain action classification is feasible. For SA classification in particular we were able to achieve very strong performance. Although performance on concept sequence and DA classification is not as high, it is still quite strong, especially given that there are an order of magnitude more classes than in SA classification. Based on our experiments, it appears that all of the learning approaches we tested were able to cope with data sparseness at the level found in our data, with the possible exception of the naïve Bayes n-gram approach (Rainbow) for the concept sequence task.

One additional point worth noting is that there is evidence that domain action classification could be performed reasonably well using only word-based information. Although our best-performing classifiers combined word and argument parse information, the naïve Bayes word bigram classifier (Rainbow) performed very well on the SA classification task. With additional data, the performance of the concept sequence and DA word bigram classifiers could be expected to improve. Cattoni et al. (2001) also apply statistical language models to DA classification. A word bigram model is trained for each DA, and the DA with the highest likelihood is assigned to each SDU. Arguments are identified using recursive transition networks, and interlingua specification constraints are used to find the most likely valid interlingua representation. Although it is clear that argument information is useful for the task, it appears that words alone can be used to achieve reasonable performance.

Another goal of our experiments was to help in the selection of a machine learning approach to be used in our hybrid analyzer. Certainly one of the most important considerations is how well

the learning approach performs the task. For SA classification, the combination of parse features and word bigram probabilities clearly gave the best performance. For concept sequence classification, no learning approach clearly outperformed any other (with the exception that the naïve Bayes n-gram approach performed worse than other approaches). However, the performance of the classifiers is not the only consideration to be made in selecting the classifier for our hybrid analyzer.

Several additional factors are also important in selecting the particular machine learning approach to be used. One important attribute of the learning approach is the speed of both classification and training. Since the classifiers are part of a translation system designed for use between two humans to facilitate (near) real-time communication, the DA classifiers must classify individual utterances online very quickly. Furthermore, since humans must write and test the argument grammars, training and batch classification should be fast so that the grammar writers can update the grammars, retrain the classifiers, and test efficiently.

The machine learning approach should also be able to easily accommodate both continuous and discrete features from a variety of sources. Possible sources for features include words and/or phrases in an utterance, the argument parse, the interlingua representation of the arguments, and properties of the dialogue (e.g. speaker tag). The classifier should be able to easily combine features from any or all of these sources.

Another desirable attribute for the machine learning approach is the ability to produce a ranked list of possible classes. Our interlingua specification defines how speech acts and concepts are allowed to combine as well as how arguments are licensed by the domain action. These constraints can be used to select an alternative DA if the best DA violates the specification.

Based on all of these considerations, the TiMBL+Rainbow classifier, which combines parse label features with word bigram probabilities, seems like an excellent choice for speech act classification. It was the most accurate classifier that we tested. Furthermore, the main TiMBL classifier meets all of the requirements discussed above except the ability to produce a complete ranked list of the classes for each instance. However, such a list could be produced as a backup from the Rainbow probability features. Adding new features to the combined classifier would also be very easy because TiMBL was the primary classifier in the combination. Finally, since both TiMBL and Rainbow provide an online server mode for classifying single instances, incorporating the combined classifier into an online translation system would not be difficult. Since there were no significant differences in the performance of most of the concept sequence classifiers, this combined approach is probably also a good option for that task.

10 Conclusion

We have described a representation of speaker intention that includes domain independent speech acts as well as domain dependent domain actions. We have shown that domain actions are a useful level of abstraction for machine translation of task-oriented dialogue, and that, in spite of their domain specificity, they are scalable to larger domains and portable to new domains.

We have also presented classifiers for domain actions that have been comparatively tested and used successfully in the NESPOLE speech-to-speech translation system. We experimentally compared the effectiveness of several machine-learning approaches for classification of domain actions, speech acts, and concept sequences on two input languages. Despite the difficulty of the classification tasks due to a large number of classes and relatively sparse data, the classifiers exhibited strong performance on all tasks. We also demonstrated how the combination of two learning approaches could be used to improve performance and overcome the weaknesses of the individual approaches.

Acknowledgements: NESPOLE was funded by NSF (Grant number 9982227) and the EU. The NESPOLE partners are ITC-irst, Universite Joseph Fourrier, Universitat Karlsruhe, APT Trentino travel board, and AETHRA telecommunications. We would like to acknowledge the contribution of the following people in particular: Fabio Pianesi, Emanuele Pianta, Nadia Mana, and Herve Blanchon.

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.

Fillmore, C.J., Kay, P. and O'Connor, M.C. 1988. Regularity and Idiomaticity in Grammatical Constructions. Language, 64(3), 501-538.

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

Goldberg, Adele E. 1995. Constructions: A Construction Grammar Approach to Argument Structure. Chicago University Press.

Langley, C. and A. Lavie. 2003. Parsing Domain Actions with Phrase-Level Grammars and Memory-Based Learners. To appear in Proceedings of IWPT-2003. Nancy, France.

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.

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

Levin, L., D. Gates, D. Wallace, K. Peterson, A. Lavie F. Pianesi, E. Pianta, R. Cattoni, N. Mana. 2002. Balancing Expressiveness and Simplicity in an Interlingua for Task Based Dialogue. In Proceedings of Workshop on Spoken Language Translation. ACL-02, Philadelphia.

McCallum, A. K. 1996. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/~mccallum/bow.

Qu, Y., B. DiEugenio, A. Lavie, L. Levin and C.P. Rose. 1997. Minimizing Cumulative Error in Discourse Context. 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.

Qu, Y., C. P. Rose, and B. DiEugenio. 1996. Using Discourse Predictions for Ambiguity Resolution. In Proceedings of COLING-1996.

Quinlan, J. R. 1993. C4.5: Programs for Machine Learning. San Mateo: Morgan Kaufmann.

Reithinger, N., R. Engel, M. Kipp, M. Klesen. 1996. Predicting Dialogue Acts for a Speech-To-Speech Translation System. DFKI GmbH Saarbruecken. Verbmobil-Report 151. http://verbmobil.dfki.de/cgi-bin/verbmobil/htbin/doc-access.cgi

Stolcke, A., K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, M. Meteer, and C. Van Ess-Dykema. 2000. Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. Computational Linguistics 26:3, 339-371.

Zell, A., G. Mamier, M. Vogt, et al. 1998. SNNS: Stuttgart Neural Network Simulator User Manual, Version 4.2.