Combining Acoustic and Pragmatic Features to Predict Recognition Performance in Spoken Dialogue Systems

Malte Gabsdil
Department of Computational Linguistics
Saarland University
Germany
gabsdil@coli.uni-sb.de

Oliver Lemon
School of Informatics
Edinburgh University
Scotland
olemon@inf.ed.ac.uk

Abstract

We use machine learners trained on a combination of acoustic confidence and pragmatic plausibility features computed from dialogue context to predict the accuracy of incoming n-best recognition hypotheses to a spoken dialogue system. Our best results show a 25% weighted f-score improvement over a baseline system that implements a "grammar-switching" approach to context-sensitive speech recognition.

1 Introduction

A crucial problem in the design of spoken dialogue systems is to decide for incoming recognition hypotheses whether a system should *accept* (consider correctly recognized), *reject* (assume misrecognition), or *ignore* (classify as noise or speech not directed to the system) them. In addition, a more sophisticated dialogue system might decide whether to *clarify* or *confirm* certain hypotheses.

Obviously, incorrect decisions at this point can have serious negative effects on system usability and user satisfaction. On the one hand, accepting misrecognized hypotheses leads to misunderstandings and unintended system behaviors which are usually difficult to recover from. On the other hand, users might get frustrated with a system that behaves too cautiously and rejects or ignores too many utterances. Thus an important feature in dialogue system engineering is the tradeoff between avoiding task failure (due to misrecognitions) and promoting overall dialogue efficiency, flow, and naturalness.

In this paper, we investigate the use of machine learners trained on a combination of acoustic confidence and pragmatic plausibility features (i.e. computed from dialogue context) to predict the quality of incoming n-best recognition hypotheses to a spoken dialogue system. These predictions are then used to select a "best" hypothesis and to decide on appropriate system reactions. We evaluate this approach in comparison with a baseline system that combines fixed recognition confidence rejection thresholds with dialogue-state dependent recognition grammars (Lemon, 2004).

The paper is organized as follows. After a short relation to previous work, Section 3 introduces the WITAS multimodal dialogue system, which we use to collect data (Section 4) and to derive baseline results (Section 5). Section 6 describes our learning experiments for classifying and selecting from n-best recognition hypotheses and Section 7 reports our results.

2 Relation to Previous Work

(Litman et al., 2000) use acoustic-prosodic information extracted from speech waveforms, together with information derived from their speech recognizer, to automatically predict misrecognized turns in a corpus of train-timetable information dialogues. In our experiments, we also use recognizer confidence scores and a limited number of acoustic-prosodic features (e.g. amplitude in the speech signal) for hypothesis classification. (Walker et al., 2000) use a combination of features from the speech recognizer, natural language understanding, and dialogue manager/discourse history to classify hypotheses as correct, partially correct, or misrecognized. Our work is related to these experiments in that we also combine confidence scores and higher-level features for classification. However, both (Litman et al., 2000) and (Walker et al., 2000) consider only single-best recognition results and thus use their classifiers as "filters" to decide whether the best recognition hypothesis for a user utterance is correct or not. We go a step further in that we classify n-best hypotheses and then select among the alternatives. We also explore the use of more dialogue and task-oriented features (e.g. the dialogue move type of a recognition hypothesis) for classification.

The main difference between our approach and work on hypothesis reordering (e.g. (Chotimongkol and Rudnicky, 2001)) is that we make a decision regarding whether a dialogue system should accept, clarify, reject, or ignore a user utterance. Furthermore, our approach is more generally applica-

ble than preceding research, since we frame our methodology in the *Information State Update* (ISU) approach to dialogue management (Traum et al., 1999) and therefore expect it to be applicable to a range of related multimodal dialogue systems.

3 The WITAS Dialogue System

The WITAS dialogue system (Lemon et al., 2002) is a multimodal command and control dialogue system that allows a human operator to interact with a simulated "unmanned aerial vehicle" (UAV): a small robotic helicopter. The human operator is provided with a GUI – an interactive (i.e. mouse clickable) map – and specifies mission goals using natural language commands spoken into a headset, or by using combinations of GUI actions and spoken commands. The simulated UAV can carry out different activities such as flying to locations, following vehicles, and delivering objects. The dialogue system uses the Nuance 8.0 speech recognizer with language models compiled from a grammar (written using the Gemini system (Dowding et al., 1993)), which is also used for parsing and generation.

3.1 WITAS Information States

The WITAS dialogue system is part of a larger family of systems that implement the *Information State Update* (ISU) approach to dialogue management (Traum et al., 1999). The ISU approach has been used to formalize different theories of dialogue and forms the basis of several dialogue system implementations in domains such as route planning, home automation, and tutorial dialogue. The ISU approach is a particularly useful testbed for our technique because it collects information relevant to dialogue context in a central data structure from which it can be easily extracted. (Lemon et al., 2002) describe in detail the components of Information States (IS) and the update procedures for processing user input and generating system responses. Here, we briefly introduce parts of the IS which are needed to understand the system's basic workings, and from which we will extract dialogue-level and task-level information for our learning experiments:

- *Dialogue Move Tree* (DMT): a tree-structure, in which each subtree of the root node represents a "thread" in the conversation, and where each node in a subtree represents an utterance made either by the system or the user. [1]
- *Active Node List* (ANL): a list that records all "active" nodes in the DMT; active nodes indicate conversational contributions that are still in some sense open, and to which new utterances can attach.
- *Activity Tree* (AT): a tree-structure representing the current, past, and planned activities that the back-end system (in this case a UAV) performs.
- *Salience List* (SL): a list of NPs introduced in the current dialogue ordered by recency.
- *Modality Buffer* (MB): a temporary store that registers click events on the GUI.

The DMT and AT are the core components of Information States. The SL and MB are subsidiary data-structures needed for interpreting and generating anaphoric expressions and definite NPs. Finally, the ANL plays a crucial role in integrating new user utterances into the DMT.

4 Data Collection

For our experiments, we use data collected in a small user study with the grammar-switching version of the WITAS dialogue system (Lemon, 2004). In this study, six subjects from Edinburgh University (4 male, 2 female) had to solve five simple tasks with the system, resulting in 30 complete dialogues.

The subjects' utterances were recorded as 8kHz 16bit waveform files and all aspects of the Information State transitions during the interactions were logged as html files. Altogether, 303 utterances were recorded in the user study ($\approx$ 10 user utterances/dialogue).

4.1 Labeling

We transcribed all user utterances and parsed the transcriptions offline using WITAS' natural language understanding component in order to get a gold-standard labeling of the data. Each utterance was labeled as either *in-grammar* or *out-of-grammar (oog)*, depending on whether its transcription could be parsed or not, or as *crosstalk*: a special marker that indicated that the input was not directed to the system (e.g. noise, laughter, self-talk, the system accidentally recording itself). For all *in-grammar* utterances we stored their interpretations (quasi-logical forms) as computed by WITAS' parser. Since the parser uses a domain-specific semantic grammar designed for this particular application, each in-grammar utterance had an interpretation that is "correct" with respect to the WITAS application.

[1] A tree is used in order to overcome the limitations of stack-based processing, see (Lemon and Gruenstein, 2004).

4.2 Simplifying Assumptions

The evaluations in the following sections make two simplifying assumptions. First, we consider a user utterance correctly recognized only if the logical form of the transcription is the same as the logical form of the recognition hypothesis. This assumption can be too strong because the system might react appropriately even if the logical forms are not literally the same. Second, if a transcribed utterance is out-of-grammar, we assume that the system cannot react appropriately. Again, this assumption might be too strong because the recognizer can accidentally map an utterance to a logical form that is equivalent to the one intended by the user.

5 The Baseline System

The baseline for our experiments is the behavior of the WITAS dialogue system that was used to collect the experimental data (using dialogue context as a predictor of language models for speech recognition, see below). We chose this baseline because it has been shown to perform significantly better than an earlier version of the system that always used the same (i.e. full) grammar for recognition (Lemon, 2004).

We evaluate the performance of the baseline by analyzing the dialogue logs from the user study. With this information, it is possible to decide how the system reacted to each user utterance. We distinguish between the following three cases:

1. *accept*: the system accepted the recognition hypothesis of a user utterance as correct.
2. *reject*: the system rejected the recognition hypothesis of a user utterance given a fixed confidence rejection threshold.
3. *ignore*: the system did not react to a user utterance at all.

These three classes map naturally to the gold-standard labels of the transcribed user utterances: the system should *accept* in-grammar utterances, *reject* out-of-grammar input, and *ignore* crosstalk.

5.1 Context-sensitive Speech Recognition

In the the WITAS dialogue system, the "grammar-switching" approach to context-sensitive speech recognition (Lemon, 2004) is implemented using the ANL. At any point in the dialogue, there is a "most active node" at the top of the ANL. The dialogue move type of this node defines the name of a language model that is used for recognizing the next user utterance. For instance, if the most active node is a system *yes-no-question* then the appropriate language model is defined by a small context-free grammar covering phrases such as "yes", "that's right", "okay", "negative", "maybe", and so on.

The WITAS dialogue system with context-sensitive speech recognition showed significantly better recognition rates than a previous version of the system that used the full grammar for recognition at all times ((Lemon, 2004) reports a 11.5% reduction in overall utterance recognition error rate). Note however that an inherent danger with grammar-switching is that the system may have wrong expectations and thus might activate a language model which is not appropriate for the user's next utterance, leading to misrecognitions or incorrect rejections.

5.2 Results

Table 1 summarizes the evaluation of the baseline system.

	System behavior		
User utterance	accept	reject	ignore
in-grammar	154/22	8	4
out-of-grammar	45	43	4
crosstalk	12	9	2

Accuracy: 65.68%
Weighted f-score: 61.81%

Table 1: WITAS dialogue system baseline results

Table 1 should be read as follows: looking at the first row, in 154 cases the system understood and accepted the correct logical form of an in-grammar utterance by the user. In 22 cases, the system accepted a logical form that differed from the one for the transcribed utterance.[2] In 8 cases, the system rejected an in-grammar utterance and in 4 cases it did not react to an in-grammar utterance at all. The second row of Table 1 shows that the system accepted 45, rejected 43, and ignored 4 user utterances whose transcriptions were out-of-grammar and could not be parsed. Finally, the third row of the table shows that the baseline system accepted 12 utterances that were not addressed to it, rejected 9, and ignored 2.

Table 1 shows that a major problem with the baseline system is that it accepts too many user utterances. In particular, the baseline system accepts the wrong interpretation for 22 in-grammar utterances, 45 utterances which it should have rejected as out-of-grammar, and 12 utterances which it should have

[2]For the computation of accuracy and weighted f-scores, these were counted as wrongly accepted out-of-grammar utterances.

ignored. All of these cases will generally lead to unintended actions by the system.

6 Classifying and Selecting N-best Recognition Hypotheses

We aim at improving over the baseline results by considering the n-best recognition hypotheses for each user utterance. Our methodology consists of two steps: i) we automatically classify the n-best recognition hypotheses for an utterance as either correctly or incorrectly recognized and ii) we use a simple selection procedure to choose the "best" hypothesis based on this classification. In order to get multiple recognition hypotheses for all utterances in the experimental data, we re-ran the speech recognizer with the full recognition grammar and 10-best output and processed the results offline with WITAS' parser, obtaining a logical form for each recognition hypothesis (every hypothesis has a logical form since language models are compiled from the parsing grammar).

6.1 Hypothesis Labeling

We labeled all hypotheses with one of the following four classes, based on the manual transcriptions of the experimental data: *in-grammar*, *oog (WER* $\leq$ *50)*, *oog (WER* $>$ *50)*, or *crosstalk*. The *in-grammar* and *crosstalk* classes correspond to those described for the baseline. However, we decided to divide up the *out-of-grammar* class into the two classes *oog (WER* $\leq$ *50)* and *oog (WER* $>$ *50)* to get a more fine-grained classification. In order to assign hypotheses to the two *oog* classes, we compute the word error rate (WER) between recognition hypotheses and the transcription of corresponding user utterances. If the WER is $\leq$ 50%, we label the hypothesis as *oog (WER* $\leq$ *50)*, otherwise as *oog (WER* $>$ *50)*. We also annotate all misrecognized hypotheses of *in-grammar* utterances with their respective WER scores.

The motivation behind splitting the *out-of-grammar* class into two subclasses and for annotating misrecognized *in-grammar* hypotheses with their WER scores is that we want to distinguish between different "degrees" of misrecognition that can be used by the dialogue system to decide whether it should initiate clarification instead of rejection.[3] We use a threshold (50%) on a hypothesis' WER as an indicator for whether hypotheses should be clarified or rejected. This is adopted from (Gabsdil, 2003), based on the fact that WER correlates with concept accuracy (CA, (Boros et al., 1996)). The WER threshold can be set differently according to the needs of an application. However, one would ideally set a threshold directly on CA scores for this labeling, but these are currently not available for our data.

We also introduce the distinction between *out-of-grammar (WER* $\leq$ *50)* and *out-of-grammar (WER* $>$ *50)* in the gold standard for the classification of (whole) user utterances. We split the *out-of-grammar* class into two sub-classes depending on whether the 10-best recognition results include at least one hypothesis with a WER ≤ 50 compared to the corresponding transcription. Thus, if there is a recognition hypothesis which is close to the transcription, an utterance is labeled as *oog (WER* $\leq$ *50)*. In order to relate these classes to different system behaviors, we define that utterances labeled as *oog (WER* $\leq$ *50)* should be *clarified* and utterances labeled as *oog (WER* $>$ *50)* should be *rejected* by the system. The same is done for all *in-grammar* utterances for which only misrecognized hypotheses are available.

6.2 Classification: Feature Groups

We represent recognition hypotheses as 20-dimensional feature vectors for automatic classification. The feature vectors combine recognizer confidence scores, low-level acoustic information, information from WITAS system Information States, and domain knowledge about the different tasks in the scenario. The following list gives an overview of all features (described in more detail below).

1. **Recognition (6):** *nbestRank, hypothesisLength, confidence, confidenceZScore, confidence-StandardDeviation, minWordConfidence*
2. **Utterance (3):** *minAmp, meanAmp, RMS-amp*
3. **Dialogue (9):** *currentDM, currentCommand, mostActiveNode, DMBigramFrequency, qaMatch, aqMatch, #unresolvedNPs, #unresolvedPronouns, #uniqueIndefinites*
4. **Task (2):** *taskConflict, #taskConstraintConflict*

All features are extracted automatically from the output of the speech recognizer, utterance waveforms, IS logs, and a small library of plan operators describing the actions the UAV can perform. The recognition (REC) feature group includes the position of a hypothesis in the n-best list (*nbestRank*),

[3] The WITAS dialogue system currently does not support this type of clarification dialogue; the WER annotations are therefore only of theoretical interest. However, an extended system could easily use this information to decide when clarification should be initiated.

its length in words (*hypothesisLength*), and five features representing the recognizer's confidence assessment. Similar features have been used in the literature (e.g. (Litman et al., 2000)). The *minWordConfidence* and standard deviation/zScore features are computed from individual word confidences in the recognition output. We expect them to help the machine learners decide between the different WER classes (e.g. a high overall confidence score can sometimes be misleading). The utterance (UTT) feature group reflects information about the amplitude in the speech signal (all features are extracted with the UNIX `sox` utility). The motivation for including the amplitude features is that they might be useful for detecting crosstalk utterances which are not directly spoken into the headset microphone (e.g. the system accidentally recognizing itself).

The dialogue features (DIAL) represent information derived from Information States and can be coarsely divided into two sub-groups. The first group includes features representing general coherence constraints on the dialogue: the dialogue move types of the current utterance (*currentDM*) and of the most active node in the ANL (*mostActiveNode*), the command type of the current utterance (*currentCommand*, if it is a command, *null* otherwise), statistics on which move types typically follow each other (*DMBigramFrequency*), and two features (*qaMatch* and *aqMatch*) that explicitly encode whether the current and the previous utterance form a valid question answer pair (e.g. *yn-question* followed by *yn-answer*). The second group includes features that indicate how many definite NPs and pronouns cannot be resolved in the current Information State (*#unresolvedNP, #unresolvedPronouns*, e.g. "the car" if no car was mentioned before) and a feature indicating the number of indefinite NPs that can be uniquely resolved in the Information State (*#uniqueIndefinites*, e.g. "a tower" where there is only one tower in the domain). We include these features because (short) determiners are often confused by speech recognizers. In the WITAS scenario, a misrecognized determiner/demonstrative pronoun can lead to confusing system behavior (e.g. a wrongly recognized "there" will cause the system to ask "Where is that?").

Finally, the task features (TASK) reflect conflicting instructions in the domain. The feature *taskConflict* indicates a conflict if the current dialogue move type is a command and that command already appears as an active task in the AT. *#taskConstraintConflict* counts the number of conflicts that arise between the currently active tasks in the AT and the hypothesis. For example, if the UAV is already flying somewhere the preconditions of the action operator for `take_off` ($altitude = 0$) conflict with those for `fly` ($altitude \neq 0$), so that "take off" would be an unlikely command in this context.

6.3 Learners and Selection Procedure

We use the memory based learner TiMBL (Daelemans et al., 2002) and the rule induction learner RIPPER (Cohen, 1995) to predict the class of each of the 10-best recognition hypotheses for a given utterance. We chose these two learners because they implement different learning strategies, are well established, fast, freely available, and easy to use. In a second step, we decide which (if any) of the classified hypotheses we actually want to pick as the best result and how the user utterance should be classified as a whole. This task is decided by the following selection procedure (see Figure 1) which implements a preference ordering *accept* $>$ *clarify* $>$ *reject* $>$ *ignore*.[4]

1. Scan the list of classified n-best recognition hypotheses top-down. Return the first result that is classified as *accept* and classify the utterance as *accept*.
2. If 1. fails, scan the list of classified n-best recognition hypotheses top-down. Return the first result that is classified as *clarify* and classify the utterance as *clarify*.
3. If 2. fails, count the number of rejects and ignores in the classified recognition hypotheses. If the number of rejects is larger or equal than the number of ignores classify the utterance as *reject*.
4. Else classify the utterance as *ignore*.

Figure 1: Selection procedure

This procedure is applied to choose from the classified n-best hypotheses for an utterance, independent of the particular machine learner, in all of the following experiments.

Since we have a limited amount experimental data in this study (10 hypotheses for each of the 303 user utterances), we use a "leave-one-out" crossvalidation setup for classification. This means that we classify the 10-best hypotheses for a particular utterance based on the 10-best hypotheses of all 302 other utterances and repeat this 303 times.

[4]Note that in a dialogue application one would not always need to classify all n-best hypotheses in order to select a result but could stop as soon as a hypothesis is classified as correct, which can save processing time.

7 Results and Evaluation

The middle part of Table 2 shows the classification results for TiMBL and RIPPER when run with default parameter settings (the other results are included for comparison). The individual rows show the performance when different combinations of feature groups are used for training. The results for the three-way classification are included for comparison with the baseline system and are obtained by combining the two classes *clarify* and *reject*. Note that we do not evaluate the performance of the learners for classifying the individual recognition hypotheses but the classification of (whole) user utterances (i.e. including the selection procedure to choose from the classified hypotheses).

The results show that both learners profit from the addition of more features concerning dialogue context and task context for classifying user speech input appropriately. The only exception from this trend is a slight performance decrease when task features are added in the four-way classification for RIPPER. Note that both learners already outperform the baseline results even when only recognition features are considered. The most striking result is the performance gain for TiMBL (almost 10%) when we include the dialogue features. As soon as dialogue features are included, TiMBL also performs slightly better than RIPPER.

Note that the introduction of (limited) task features, in addition to the DIAL and UTT features, did not have dramatic impact in this study. One aim for future work is to define and analyze the influence of further task related features for classification.

7.1 Optimizing TiMBL Parameters

In all of the above experiments we ran the machine learners with their default parameter settings. However, recent research (Daelemans and Hoste, 2002; Marsi et al., 2003) has shown that machine learners often profit from parameter optimization (i.e. finding the best performing parameters on some development data). We therefore selected 40 possible parameter combinations for TiMBL (varying the number of nearest neighbors, feature weighting, and class voting weights) and nested a parameter optimization step into the "leave-one-out" evaluation paradigm (cf. Figure 2).[5]

Note that our optimization method is not as sophisticated as the "Iterative Deepening" approach described by (Marsi et al., 2003) but is similar in the sense that it computes a best-performing parameter setting for each data fold.

1. Set aside the recognition hypotheses for one of the user utterances.
2. Randomly split the remaining data into an 80% training and 20% test set.
3. Run TiMBL with all possible parameter settings on the generated training and test sets and store the best performing settings.
4. Classify the left-out hypotheses with the recorded parameter settings.
5. Iterate.

Figure 2: Parameter optimization

Table 3 shows the classification results when we run TiMBL with optimized parameter settings and using all feature groups for training.

	System Behavior			
User Utterance	accept	clarify	reject	ignore
in-grammar	159/2	11	16	0
out-of-grammar (WER $\leq$ 50)	0	25	5	0
out-of-grammar (WER $>$ 50)	6	6	50	0
crosstalk	2	5	0	16

Acc/wf-score (3 classes): 86.14/86.39%
Acc/wf-score (4 classes): 82.51/83.29%

Table 3: TiMBL classification results with optimized parameters

Table 3 shows a remarkable 9% improvement for the 3-way and 4-way classification in both accuracy and weighted f-score, compared to using TiMBL with default parameter settings. In terms of WER, the baseline system (cf. Table 1) accepted 233 user utterances with a WER of 21.51%, and in contrast, TiMBL with optimized parameters (Ti_OP) only accepted 169 user utterances with a WER of 4.05%. This low WER reflects the fact that if the machine learning system accepts an user utterance, it is almost certainly the correct one. Note that although the machine learning system in total accepted far fewer utterances (169 vs. 233) it accepted more correct utterances than the baseline (159 vs. 154).

7.2 Evaluation

The baseline accuracy for the 3-class problem is 65.68% (61.81% weighted f-score). Our best results, obtained by using TiMBL with parameter op-

[5] We only optimized parameters for TiMBL because it performed better with default settings than RIPPER and because the findings in (Daelemans and Hoste, 2002) indicate that TiMBL profits more from parameter optimization.

System or features used for classification	Acc/wf-score (3 classes)	Acc/wf-score (4 classes)	Acc/wf-score (3 classes)	Acc/wf-score (4 classes)
Baseline	65.68/61.81%			
	TiMBL		RIPPER	
REC	67.66/67.51%	63.04/63.03%	69.31/69.03%	66.67/65.14%
REC+UTT	68.98/68.32%	64.03/63.08%	72.61/72.33%	70.30/68.61%
REC+UTT+DIAL	77.56/77.59%	72.94/73.70%	74.92/75.34%	71.29/71.62%
REC+UTT+DIAL+TASK	77.89/77.91%	73.27/74.12%	75.25/75.61%	70.63/71.54%
TiMBL (optimized params.)	86.14/86.39%	82.51/83.29%		
Oracle	94.06/94.17%	94.06/94.18%		

Table 2: Classification Results

timization, show a 25% weighted f-score improvement over the baseline system.

We can compare these results to a hypothetical "oracle" system in order to obtain an upper bound on classification performance. This is an imaginary system which performs perfectly on the experimental data given the 10-best recognition output. The oracle results reveal that for 18 of the in-grammar utterances the 10-best recognition hypotheses do not include the correct logical form at all and therefore have to be classified as *clarify* or *reject* (i.e. it is not possible to achieve 100% accuracy on the experimental data). Table 2 shows that our best results are only 8%/12% (absolute) away from the optimal performance.

7.2.1 Costs and χ^2 Levels of Significance

We use the χ^2 test of independence to statistically compare the different classification results. However, since χ^2 only tells us whether two classifications are different from each other, we introduce a simple cost measure (Table 4) for the 3-way classification problem to complement the χ^2 results.[6]

	System behavior		
User utterance	accept	reject	ignore
in-grammar	0	2	2
out-of-grammar	4	2	2
crosstalk	4	2	0

Table 4: Cost measure

Table 4 captures the intuition that the correct behavior of a dialogue system is to accept correctly recognized utterances and ignore crosstalk (cost 0). The worst a system can do is to accept misrecognized utterances or utterances that were not addressed to the system. The remaining classes are assigned a value in-between these two extremes. Note that the cost assignment is not validated against user judgments. We only use the costs to interpret the χ^2 levels of significance (i.e. as an indicator to compare the relative quality of different systems).

Table 5 shows the differences in cost and χ^2 levels of significance when we compare the classification results. Here, Ti_OP stands for TiMBL with optimized parameters and the stars indicate the level of statistical significance as computed by the χ^2 statistics (*** indicates significance at $p = .001$, ** at $p = .01$, and * at $p = .05$).[7]

	Baseline	RIPPER	TiMBL	Ti_OP
Oracle	-232^{***}	-116^{***}	-100^{***}	-56
Ti_OP	-176^{***}	-60^{*}	-44	
TiMBL	-132^{***}	-16		
RIPPER	-116^{***}			

Table 5: Cost comparisons and χ^2 levels of significance for 3-way classification

The cost measure shows the strict ordering: Oracle $<$ Ti_OP $<$ TiMBL $<$ RIPPER $<$ Baseline. Note however that according to the χ^2 test there is no significant difference between the oracle system and TiMBL with optimized parameters. Table 5 also shows that all of our experiments significantly outperform the baseline system.

8 Conclusion

We used a combination of acoustic confidence and pragmatic plausibility features (i.e. computed from dialogue context) to predict the quality of incoming recognition hypotheses to a multi-modal dialogue system. We classified hypotheses as *accept*, (*clarify*), *reject*, or *ignore*: functional categories that

[6] We only evaluate the 3-way classification problem because there are no baseline results for the 4-way classification available.

[7] Following (Hinton, 1995), we leave out categories with expected frequencies < 5 in the χ^2 computation and reduce the degrees of freedom accordingly.

can be used by a dialogue manager to decide appropriate system reactions. The approach is novel in combining machine learning with n-best processing for spoken dialogue systems using the Information State Update approach.

Our best results, obtained using TiMBL with optimized parameters, show a 25% weighted f-score improvement over a baseline system that uses a "grammar-switching" approach to context-sensitive speech recognition, and are only 8% away from the optimal performance that can be achieved on the data. Clearly, this improvement would result in better dialogue system performance overall. Parameter optimization improved the classification results by 9% compared to using the learner with default settings, which shows the importance of such tuning.

Future work points in two directions: first, integrating our methodology into working ISU-based dialogue systems and determining whether or not they improve in terms of standard dialogue evaluation metrics (e.g. task completion). The ISU approach is a particularly useful testbed for our methodology because it collects information pertaining to dialogue context in a central data structure from which it can be easily extracted. This avenue will be further explored in the TALK project[8]. Second, it will be interesting to investigate the impact of different dialogue and task features for classification and to introduce a distinction between "generic" features that are domain independent and "application-specific" features which reflect properties of individual systems and application scenarios.

Acknowledgments

We thank Nuance Communications Inc. for the use of their speech recognition and synthesis software and Alexander Koller and Dan Shapiro for reading draft versions of this paper. Oliver Lemon was partially supported by Scottish Enterprise under the Edinburgh-Stanford Link programme.

References

M. Boros, W. Eckert, F. Gallwitz, G. Görz, G. Hanrieder, and H. Niemann. 1996. Towards Understanding Spontaneous Speech: Word Accuracy vs. Concept Accuracy. In *Proc. ICSLP-96*.

Ananlada Chotimongkol and Alexander I. Rudnicky. 2001. N-best Speech Hypotheses Reordering Using Linear Regression. In *Proceedings of EuroSpeech 2001*, pages 1829–1832.

William W. Cohen. 1995. Fast Effective Rule Induction. In *Proceedings of the 12th International Conference on Machine Learning*.

Walter Daelemans and Véronique Hoste. 2002. Evaluation of Machine Learning Methods for Natural Language Processing Tasks. In *Proceedings of LREC-02*.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2002. TIMBL: Tilburg Memory Based Learner, version 4.2, Reference Guide. In *ILK Technical Report 02-01*.

John Dowding, Jean Mark Gawron, Doug Appelt, John Bear, Lynn Cherny, Robert Moore, and Douglas Moran. 1993. GEMINI: a natural language system for spoken-language understanding. In *Proceedings of ACL-93*.

Malte Gabsdil. 2003. Classifying Recognition Results for Spoken Dialogue Systems. In *Proceedings of the Student Research Workshop at ACL-03*.

Perry R. Hinton. 1995. *Statistics Explained – A Guide For Social Science Students*. Routledge.

Oliver Lemon and Alexander Gruenstein. 2004. Multithreaded context for robust conversational interfaces: context-sensitive speech recognition and interpretation of corrective fragments. *ACM Transactions on Computer-Human Interaction*. (to appear).

Oliver Lemon, Alexander Gruenstein, and Stanley Peters. 2002. Collaborative activities and multitasking in dialogue systems. *Traitement Automatique des Langues*, 43(2):131–154.

Oliver Lemon. 2004. Context-sensitive speech recognition in ISU dialogue systems: results for the grammar switching approach. In *Proceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue, CATALOG'04*.

Diane J. Litman, Julia Hirschberg, and Marc Swerts. 2000. Predicting Automatic Speech Recognition Performance Using Prosodic Cues. In *Proceedings of NAACL-00*.

Erwin Marsi, Martin Reynaert, Antal van den Bosch, Walter Daelemans, and Véronique Hoste. 2003. Learning to predict pitch accents and prosodic boundaries in Dutch. In *Proceedings of ACL-03*.

David Traum, Johan Bos, Robin Cooper, Staffan Larsson, Ian Lewin, Colin Matheson, and Massimo Poesio. 1999. A Model of Dialogue Moves and Information State Revision. Technical Report D2.1, Trindi Project.

Marilyn Walker, Jerry Wright, and Irene Langkilde. 2000. Using Natural Language Processing and Discourse Features to Identify Understanding Errors in a Spoken Dialogue System. In *Proceedings of ICML-2000*.

[8]EC FP6 IST-507802, http://www.talk-project.org