Multi-feature Error Detection in Spoken Dialogue Systems

Piroska Lendvai[∗], *Antal van den Bosch*[∗], *Emiel Krahmer*[∗], *Marc Swerts*[†]

∗ Computational Linguistics and AI, Tilburg University
† TU/e / CNTS, Eindhoven University of Technology / Antwerp University

Abstract

The present paper evaluates the role selected features and feature combinations play for error detection in spoken dialogue systems. We investigate the relevance of various, readily available features extracted from a corpus of dialogues with a train timetable information system, using RIPPER, a rule-inducing machine learning algorithm. The learning task consists of the identification of communication problems arising in either the previous turn or the current turn of the dialogue. Previous experiments with our corpus have shown that combining dialogue history and word-graph features is beneficial for detecting errors (in particular in the previous turn). Other researchers have reported that combining prosodic and ASR characteristics is helpful (primarily in the current turn). In this paper, we investigate the usefulness of large-scale combinations of these features for the above two tasks. We show that we are unable to reproduce the benefits of prosodic features for learning problematic situations, even though the overall prosodic trends in our corpus are similar to those earlier reported on. Moreover, the best results are obtained using just minimal combinations of two sources of information.

1 Introduction

There is increasing interest in using machine learning (ML) for automatic error detection in spoken dialogue systems (SDS). Such studies are largely varied with respect to their definitions of the problem detection task, the learning methods applied, and the attributes employed in the learning method. A wide spectrum of features is exploited in such experiments, depending on what sources of information are regarded to be relevant in the underlying task. Features are selected on grounds of their supposed predictive power towards problems during the interaction between the system and its user. The employed features range from primitive attributes representing entities such as confidence scores output by the automatic speech recognition (ASR) module of the system (Hirschberg, Litman, and Swerts, 1999; Litman, Walker, and Kearns, 1999), lexical output of the ASR module of a SDS (Hirschberg, Litman, and Swerts, 1999; Van den Bosch, Krahmer, and Swerts, 2001), experimental parameters and identification of the underlying ASR grammar (Hirschberg, Litman, and Swerts, 1999; Litman, Walker, and Kearns, 1999), aspects of dialogue efficiency and quality (Litman, Walker, and Kearns, 1999), presence or absence of default assumptions, the amount of slots filled (Krahmer et al., 1999) or the system adaptivity (Hirschberg, Litman, and Swerts, 1999), to highly complicated features, involving a variety of semantics-based attributes of the user input (Hirschberg, Litman, and Swerts, 1999; Litman, Walker, and Kearns, 1999; Walker, Wright, and Langkilde, 2000), and aspects of syntax in the user answer (Krahmer et al., 1999). As opposed to the primitive attributes, the latter types of features cannot be straightforwardly extracted from a SDS, which forms an obstacle for automatic, on-line error detection and recovery.

Recently there is an emerging trend to incorporate prosody, defined as the set of suprasegmental speech features, such as intonation (speech melody), tempo, pausing and loudness, into the modules of automatic speech recognition and understanding of diverse applications. The various tasks include the attempted use of prosodic structures as a pre- or a postprocessor, e.g., to rerank n-best lists of recognition hypotheses (Veilleux and Ostendorf, 1994; Hirose, 1995); to run separate models for words that are or are not accented (Greenberg, 2001); to automatically punctuate transcribed spoken texts; to chunk a continuous stream of speech into smaller parts before it is fed into the recognition module and to classify speaker turns in terms of a set of dialogue acts to run act-specific language models (Taylor et al., 1996). More recently, people have started to explore whether prosody may also be useful as a resource for error detection. Current SDS still make lots of errors when they have to recognize spoken input from users, so the dialogue manager (DM) of such systems needs a principled strategy to decide when it can ‘believe’ a certain recognized string. Traditional reliability measures, including acoustic and semantic confidence scores, are still not efficient enough. The reason to investigate prosody for the purpose of error detection is partly motivated by the fact that it functions eminently well as a cue to problems in human-human interactions, e.g. (Shimojima et al., 1998). However, results from previous studies on prosodically based error detection tools in SDS are a bit inconclusive, since they appear to work well for some systems (Hirschberg, Litman, and Swerts, 1999), but are far less successful for others (Lendvai et al., 2002). As it is largely unknown why there is so much variance in performance of these different tools, the current study aims to gain further insight into the relative importance of prosody for error detection on the basis of a series of experiments.

In addition, from a machine learning perspective, it is interesting to learn how prosodic features that are essentially continuous in nature, combine with more discrete, categorical (symbolic) features, such as a word graph output by the ASR or various aspects of the dialogue system, which have also been investigated in terms of their usefulness for error detection. It is still an open empirical question what machine learning approach is best capable of integrating multiple features of widely different nature.

The current study explicitly investigates the feasibility and the usefulness of combining different knowledge sources for error detection tasks. In doing such a multi-feature error detection exercise, we will treat this classification problem not as one general task applied to a full set of recorded dialogues, but rather run experiments that take into account what the specific dialogue situation is in which a problem has occurred. That is, while different investigations have brought to light that prosodic behaviour in relation to dialogue problems very much depends on the specific situation at hand (Litman and Pan, 1999; Swerts, Litman, and Hirschberg, 2000), this fact has not yet fully been exploited in error detection tasks. We will propose to do this by looking at the type of question the dialogue system has posed to the user most recently in the course of the interaction.

The paper's structure is the following. In Section 2 we first describe the dialogue data used for our study and the schema used for annotating communication problems. Subsequently, we describe the collection of features hypothesized to be predictive in detecting communication problems. In Section 3, the learning tasks and the learning method are described. Experimental results on the different feature types are treated in Section 4, after which our conclusions are given in Section 5.

2 Data

2.1 Corpus and labeling

The corpus used in the present study consists of 441 full dialogues, broken down into 3738 pairs of system questions and user answers that were sampled from a range of telephone calls of users with a Dutch human-machine train information system. Virtually all dialogues involve a different speaker. The SDS prompts the user for information needed to perform a train timetable database query. It gives feedback on what it has understood from the user input via implicit or explicit verification, thus the user will always become aware of eventual misunderstandings from the following system question. The percentage of unsuccessful conversations in our corpus is 47.6%. Problems emerge primarily because of poor speech recognition and ineffective dialogue management, and secondarily because of erroneous user inputs or false default assumptions by the system. The errors were annotated by three persons. All data were annotated by at least two of them, and differences in annotation were resolved through discussion. Labeling marks whether or not the user's input gave rise to communication problems in the course of the conversation. This feature is the one to be predicted by the machine learner. Further annotation of the dialogues concerns the type of prompt the system has most recently given. The types of system question are the following: Open question (O), Explicit verification (E), Implicit verification (I), Yes/no question (Y), Meta-question (M), and eXceptional behavior (X), if necessary in combination with the suffix Repetition (R) which indicates that the current system prompt is a repetition of the previous prompt.

To illustrate the labeling task, consider Figure 1 containing the first three pairs of system questions and user answers in a running dialogue, where "S" denotes a system question, and "U" the user answer. Apparently, the first user utterance is not recognised correctly; the second system question is about the departure and arrival stations, which the user has just given. However, the unsolicited information about the day of travel is correctly understood from the user input and is verified implicitly in the prompt. Thus, our labeling marks the first user turn as "PROBLEM", meaning that processing this utterance caused some communication problem in the dialogue.

Turn #	S/U	Utterance	Annotation
1	S	Goedenavond(. . .). Van welk station naar welk station wilt u reizen? *(Good evening. From which station to which station do you want to travel?)*	O
	U	Ik moet volgende week dinsdag van Schiphol naar Nijmegen. (*I need to go from Schiphol to Nijmegen on Tuesday next week.)*	PROBLEM
2	S	Van waar naar waar wilt u op dinsdag twaalf december reizen? (*From where to where would you like to travel on Tuesday twelve December?*)	I
	U	Van Schiphol naar Nijmegen. (*From Schiphol to Nijmegen.*)	NO PROBLEM
3	S	Hoe laat wilt u vanuit Schiphol naar Nijmegen reizen? (*At what time do you want to travel from Schiphol to Nijmegen?*)	I
	U	Rond kwart over elf 's avonds. (*Around quarter past eleven in the evening.*)	PROBLEM

Figure 1: The first three turns of an example dialogue.

2.2 Feature representations

The primary hypothesis underlying our study is that communication problems have concrete correlates in what the user is saying at a certain point of the dialogue. The relevant attributes should be selected, and training data should be represented according to these features in order to train a ML algorithm on the problem detection task. We designed a conversion step of the SDS data to instances, where one instance represents a "current" state in the dialogue system. The selected features that make up these representations are deliberately low-level; moreover, they can all be automatically extracted in real time from the online system, of which the internal states were available to us in logs and audio files. Table 1 lists the dialogue characteristics used. We distilled features both from the state of the system and from what is recognised from the user's reply.

From the user, we use both the output of the ASR module and the raw audio. The ASR output of this particular system produced a word graph, from which we stripped all recognised words, encoding these in total as a 759-bit binary bag-of-words vector. The 759 bits represent all words that occurred in our corpus. This bag-of-words representation originates from the vector space model for document representation, used in information retrieval (Salton, 1989). Further user features extracted from the word graph are the duration of the initial pause, the speech tempo, and the degree of branching in the word graph. Another lexical attribute used is the most confidently recognized string in the word graph.

The initial pause in the utterance (the length of the silence that precedes the utterance) is assumed to indicate the degree of hesitation of the user in responding,

Aspect	Feature
DM: prompt	six previous question types
ASR: confidence	summed confidence score of most confident path
ASR: branching	branching factor in the word graph of current and previous utterance
ASR: lexical	bag-of-words of previous and current user turn; most confident recognized string
Prosody: pitch	maximum and minimum F0; position of maximum and minimum; mean F0 and standard deviation
Prosody: energy	maximum energy (RMS); position of maximum; mean RMS and standard deviation
Prosody: duration	length of utterance in seconds; length of initial pause in frames
Prosody: tempo	number of syllables per second

Table 1: Overview of the employed features

cf. (Krahmer et al., 2001). The speech tempo of the utterance corresponds to the number of uttered syllables per second. To compute the number of syllables in an utterance, we used a memory-based syllabifier for Dutch (Daelemans and van den Bosch, 1992). The branching factor in the ASR word graph was also calculated both for the current and the preceding utterance, characterizing a degree of confusion in the graph; a lot of branching in the word graph can be an indicator of system uncertainty, or noisy user input. The confidence measurements of the ASR were also converted into a feature: we used the total confidence, summed over nodes, of the overall most confident path.

From the audio we automatically extracted F0 (fundamental frequency) measurements, RMS (energy, amplitude) measurements and duration of the utterance from initial silence to final silence, using the GIPOS software package. The method used to determine F0 is Hermes' method of subharmonic summation (Hermes, 1988) combined with dynamic programming to smooth the F0 contour and remove any possible pitch measuring errors.

To conclude our feature selection, we selected the sequence of the six most recent system question types as a superficial representation of the dialogue so far. The number of six is arbitrarily chosen; the assumption is that some patterns in the sequences of questions may mark typically problematic situations at the next utterance (such as two or more repetitions of the same question type), but it is unlikely that essential parts of these patterns will originate in the questions asked five or more turns before.

2.3 Descriptive Statistics

A statistical description of the prosodic properties of our data is given in tables 2 and 3 that show the mean values of prosodic features calculated from all non-problematic and all problematic utterances in the corpus. We performed a paired

	Qt = E			Qt = I			Qt = O			Qt = Y		
Feature	¬Pr		Pr	¬Pr		Pr	¬Pr		Pr	¬Pr		Pr
F0 max *(Hz)*	217.8		224.2	236.0		237.6	241.6		238.8	213.3		207.6
F0 mean *(Hz)*	144.4		148.4	159.9		161.7	160.7		163.8	147.5		150.7
RMS max	5507.2		5421.5	6683.7	**	5540.7	7325.9	**	6091.0	5173.7	**	4290.0
RMS mean	241.8		264.9	378.3	**	330.5	447.2	**	379.6	248.1		240.8
Duration *(s)*	1.9	**	2.6	2.7	**	2.9	3.6	**	3.4	1.8	*	2.0
Tempo *(syll/s)*	1.0		1.4	2.0		2.1	2.4	*	2.5	1.0	**	1.3

Table 2: Prosodic means of unproblematic (¬Pr) and problematic (Pr) current turns for four system question types ("Qt"). "*" denotes statistical differences between the two means in a paired t-test with $p < .05$ significance; "**" denotes $p < .01$ significance.

t-test on these pairs of means to check whether the differences between them are of statistical significance. The utterances are, furthermore, grouped according to their co-occurence with the four most common types of system prompts during the dialogue, characterizing the utterances per prompt type (Explicit verification, Implicit verification, Open question, and Yes/no question). Figures for the remaining prompt types are not included in the table; they occur less frequently and produce fewer statistically significant outcomes. Figures are not given for all prosodic attributes, but typically the other F0 and RMS measurements correlate statistically, as calculated in a Pearson's correlation test, thus for example a high F0 maximum often is accompanied by a high F0 mean measurement.

If we compare the actual values of the means for problematic turns according to the most recently asked system question (Qt), we find that these vary across prompt types: for example, the mean length of problematic answers following a Yes/no question is generally much shorter than the mean length of problematic answers following an Explicit verification question (cf. Table 2). Table 3 reveals such subtilities that a user's answer following an Implicit verification of a misunderstood information does not tend to be spoken louder, as one would expect (Hirschberg, Litman, and Swerts, 2000; Oviatt, McEachern, and Levow, 1998) as a consequence of hyperarticulation. Judged by the outcomes of the t-test, characteristics of some of the prosodic attributes are in accordance with findings concerning hyperarticulate speech, but others are clearly not, when distinguishing according to the actual prompt type. Furthermore, the t-test reveals that the scales of the differences between means of problematic and unproblematic turns depend on whether the communication problem occurs in the current turn (Table 2) or in the previous turn (Table 3) of the dialogue. What follows from Tables 2 and 3 is that it may be useful to decompose the error detection task into two tasks, and that the type of system prompt just given may hold predictive power towards detecting problems. The importance of the system question type was exploited in our machine learning method, as set out in the following subsection.

	Qt = E			Qt = I			Qt = O			Qt = Y		
Feature	¬Pr		Pr	¬Pr		Pr	¬Pr		Pr	¬Pr		Pr
F0 max *(Hz)*	209.8	**	236.5	234.4		240.8	236.6	**	272.7	211.8	*	263.8
F0 mean *(Hz)*	142.1	*	151.3	159.2		163.6	161.4	*	175.1	147.7	*	181.7
RMS max	5050.6	*	6253.5	6133.3		5919.7	6637.5		5936.6	5010.0		5189.3
RMS mean	206.9	**	320.2	348.7		356.5	411.3	*	353.7	246.4		295.2
Duration *(s)*	1.8	**	2.7	2.6	**	3.1	3.5		3.3	1.8	**	2.6
Tempo *(syll/s)*	0.9	**	1.4	2.0		2.1	2.5	**	1.9	1.0	**	1.6

Table 3: Prosodic means of the current turn, depending on whether the previous turn was unproblematic (¬Pr) or problematic (Pr), given for four system question types ("Qt"). "*" denotes statistical differences between the two means in a paired t-test with $p < .05$ significance; "**" denotes $p < .01$ significance.

3 Machine Learning applied to automatic error detection

3.1 Task specification

Drawing on the statistical characteristics of our corpus, error detection is carried out by means of two different series of experiments: (1) predicting miscommunication in the current user utterance versus (2) detecting a miscommunication problem in the previous user utterance given the most recent question-answer pair. Predicting whether the current user utterance will cause problems (henceforth: current-turn-problem, CTP) has been reported being more difficult (Van den Bosch, Krahmer, and Swerts, 2001; Lendvai et al., 2002), since this task has not only to deal with problems that are due to cognitive misunderstandings between the two parties, such as assumptions and presuppositions, but also to filter out technical factors that pose problems to the given dialogue system itself, such as its inability to cope with hyperarticulation or with noisy input.

The second task, aimed at identifying problems that emerged in the previous turn of the dialogue (henceforth: previous-turn-problem, PTP), consists of spotting turns signaling that the processing of the previous user input went wrong. The classifier of the PTP task can thus draw additional information from the subsequent, aware turn of the user; cf. (Litman, Hirschberg, and Swerts, 2001), where people give feedback about the progress of the communication by means of prosody (Hirschberg, Litman, and Swerts, 2000) and by means of implicit and explicit lexical cues (Van den Bosch, Krahmer, and Swerts, 2001; Krahmer et al., 1999).

It is important to distinguish between these two tasks because in this way we have a two-fold approach to error detection in SDS. Note also that there can be different labels assigned to the same feature values across the two tasks, as certain utterances are unproblematic in the current turn (CTP task) but at the same time reflect awareness of problems that occurred in the previous turn of the dialogue (PTP task). By differentiating between the two tasks, thus training separate classifiers on the tasks, we reuse the data in a unified, but still double-perspective way of error detection, enabling classification of subtle processes within one utterance.

For illustration, compare again the respective values in Table 2 and Table 3 for discovering the dissimilarity between actual feature values in the two tasks. The figures in the tables are rather different not only within one dialogue attribute (across a row), but also with respect to the same attribute across the two tasks. For example, in the learning task of the CTP, the classifier might use the information that problematic utterances tend to be produced with a faster speech tempo after an Open question, whereas utterances that signal a miscommunication in the preceding turn (the PTP task) are produced with a slower tempo than the unproblematic ones.

3.2 Learning method

In a previous study (Lendvai et al., 2002) we employed a rule induction method with domain knowledge incorporated as enforced conditioning on the induced rules, the knowledge being the type of question the system has asked in its most recent prompt. In other words, all induced rules condition at least on a question type value, in combination with zero or more conditions on other features. This approach enhanced learner performance on all attributes: in the previous-turn-task it resulted in an average 25% improvement of learning accuracy in identifying errors, indeed indicating that the scale of difference between the prosodic means is correlated with the type of system question to which users respond.

In the current study we use RIPPER (Cohen, 1995) to automatically perform error detection, based on the above method. RIPPER is a fast rule induction algorithm that induces a ruleset based on the training examples. It first separates the training set in two, then on the basis of one part it induces rules, heuristically maximizing coverage and accuracy for each rule, with potential overfitting. When the induced rules classify instances in the other part below a certain threshold, they are not stored. Rules are induced per class, ordered from low-frequency classes to high-frequency ones, leaving the most frequent class as the default rule, which is generally beneficial for the size of the rule set. RIPPER was used with its standard settings[1].

During the experiments training and testing was done by 10-fold cross-validation, where partitioning was done with complete dialogues as units, thereby ensuring that no material from the same dialogue could be part of both the training and the test set. The performance of the classifier was evaluated according to measures of predictive accuracy on deciding between problematic and unproblematic instances, and precision, recall, and F-score of the correct detection of errors. The latter metric combines precision and recall in a single figure. We employ the unweighted variant of F-score, which is defined as $2PR/(P+R)$ (P = precision, R = recall) (van Rijsbergen, 1979). In evaluating a classifier's performance in error detection more importance should be given to values of F-score than to predictive accuracy as the given F-score characterizes the rate of precision and recall for

[1] We used RIPPER version 1, release 2.4.

the prediction of the problem class while accuracy can be opaquely biased to the majority non-problem class.

3.3 Baselines

Because of the inherent differences in the two prediction tasks, two different baselines were established. For predicting miscommunication in the current turn of the dialogue, a majority-class baseline is calculated.When the data are split according to the last system question type, some question types are followed by more problematic utterances than unproblematic ones, such as open questions (“O”), repeated open questions (“OR”), and implicit verification questions (“I”). Always guessing the majority class given the prompt type produces a baseline of 65.2% accuracy and an F-score of 62.4%.

For the task of identifying a communication problem in the previous turn of the conversation, we make use of the fact that the system is already aware of a lot of problems; this is signalled directly whenever the system repeats its previous prompt. Applying a strategy of always identifying a problem when the last system prompt is repeated gives a higher baseline for the second task, henceforth referred to as the “system knows” baseline. There are 974 of these questions in the corpus, yielding 82.9% accuracy and 75.3% F-score.

4 Results

We describe the results in three steps. First, we investigate the predictive power of prosodic features in detecting communication problems in the current and the previous utterance, to test the hypothesis that prosody offers concrete correlates with problems. Second, we discuss the results on the same tasks using all non-prosodic features from the ASR wordgraph and the system questions. Third, we review the results obtained with combinations of both types of features.

4.1 Prosodic features in the error detection task

As indicated in the introductory section, some studies claim that prosody offers strong clues in automatic error detection. We tested this claim for our Dutch data by creating a matrix of combinations of prosodic features as input to the CTP and PTP tasks. The results of our experiments show no clear-cut differences between performance of prosodic features in isolation or in combinations. For the CTP task, no features or combinations could even outperform the combined-majority-class baseline, as can be seen in the left half of Table 4. This table illustrates the performance (in terms of accuracy and F-score) of only those prosodic features that outperform the baseline for the PTP task (right column). We see that for the PTP task at least some prosodic features show a significant improvement: more than 20% error-reduction in terms of F-score for duration and the set of all prosodic features produces a fair result.

It is worth noting that duration has proven to be an overall well-performing feature in the course of the experiments, isolated as well as in combination with other

	Current Turn / CTP		Previous Turn / PTP	
feature set	acc.	F	acc.	F
Baseline	65.2	62.4	82.9	75.3
Qt + F0 mean	64.7±1.4	60.6±2.6	82.4±2.0	76.5±2.9
Qt + F0 minpos	63.9±2.1	60.5±2.3	84.7±1.9	79.7±2.9
Qt + RMS maxpos	63.9±1.9	59.4±2.9	84.3±2.2	79.4±3.1
Qt + RMS mean	64.1±2.3	57.9±3.6	83.3±1.8	78.5±3.0
Qt + Duration	64.2±1.4	58.4±4.6	85.2±1.6	81.1±2.5
Qt + Tempo	64.8±2.4	59.0±3.5	84.2±1.6	78.4±3.0
Qt + All Prosodic	64.1±2.2	57.3±2.2	84.7±2.2	80.8±2.8

Table 4: Most prominent test performances in terms of accuracy and F-score trained on detecting miscommunication originating from the current or the previous turn, based on prosodic features

prosodic features. This corresponds to reports of (Batliner et al., 2001; Hirschberg, Litman, and Swerts, 1999). In the latter study duration performs with an error rate of 17.1% (corresponding to 82.9% accuracy), which is comparable to the 85.2% accuracy that our classifier gives.

4.2 Non-prosodic features in the error detection task

Table 5 lists the accuracies and F-scores of both error detection tasks based on (combinations of) non-prosodic features, viz. those features extracted from the word graph that outperform the “system-knows” baseline (confidence, branching factor, bag-of-words vector, most confident string), and the system history (the five previously asked question types, 5Q).

The CTP task is performed only slightly above baseline by the history of five before-previous questions, as well as by the combination of all non-prosodic features; recall that none of the prosodic features helped to reach the baseline (cf. Table 4). The baseline for the PTP task, however, is beaten by almost all the non-prosodic features; only slightly by the wordgraph confidence and branching factor features, but largely beaten by the bag-of-words vector and all other tested combinations of non-prosodic features that include the bag-of-words vector, all leading to an accuracy of about 91% and an F-score of about 89%.

It is worth paying attention to the lexical features, namely to the two sets of bag-of-words (BoW) and the most confident string in the word graph (ASR string). Other studies (Hirschberg, Litman, and Swerts, 1999; Litman, Walker, and Kearns, 1999) reported that the ASR string is highly relevant in predicting recognition errors (which partly corresponds to our CTP task). In the study of (Hirschberg, Litman, and Swerts, 1999) the recognized string was the best performing isolated feature, yielding an error rate of 14.4% (85.6% accuracy). This score is much higher than the result that we get (65.3% accuracy),thus we cannot regard the most confident ASR string as a well-performing feature for our goal. Note that

	Current Turn / CTP		Previous Turn / PTP	
feature set	acc.	F	acc.	F
Baseline	65.2	62.4	82.9	75.3
Q*t* + Confidence	63.9±1.8	58.0±4.0	84.9±2.0	80.5±2.8
Q*t* + BF	66.3±3.1	58.9±5.1	84.8±1.8	81.2±2.9
Q*t* + ASR string	65.3±1.8	62.2±2.2	83.6±1.5	77.0±1.9
Q*t* + BoW	66.3±1.9	61.7±3.1	90.8±1.8	88.7±2.2
Q*t* + 5Q	66.9±3.5	64.3±4.9	83.6±1.4	76.9±2.7
Q*t* + 5Q + BoW	67.7±1.6	63.2±3.1	91.1±1.1	89.4±1.3
Q*t* + 5Q + BoW + BF	69.1±2.5	64.0±3.2	90.8±1.2	89.0±1.2
Q*t* + BoW + BF	66.9±2.6	60.8±3.6	90.9±1.1	89.1±1.3
Q*t* + All Nonprosodic	69.3±2.8	65.0±3.9	91.0±0.9	89.2±1.3

Table 5: Most prominent test performances (accuracy and F-score) trained on the CTP and PTP tasks, based on non-prosodic features. BF stands for branching factor; BoW stands for bag-of-words vector; 5Q stands for the five before-previous system question types.

(Hirschberg, Litman, and Swerts, 1999) are dubious about the potential benefits of using the ASR string in error detection, questioning whether the model learned training on this feature can generalize across systems or tasks. Our experiments have indeed shown this discrepancy.

(Litman, Walker, and Kearns, 1999) employ the ASR text feature as a set-valued lexical feature in RIPPER where it also turns out to be the most predictive feature in isolation (72% accuracy) for detecting poor speech recognition. It is noteworthy that for our task the ASR string feature is less beneficial. However, a feature of the same type, the set of bag-of-words is undoubtedly the ultimate winner in our experiment matrix. An interesting question arising from this is whether this gain in favor of the BoW feature originates in the encoding differences (set-valued versus binary representation) or the fact that the BoW vector tends to contain at least parts of what the user actually said (along with all misrecognised alternatives), whereas the ASR string can be completely incorrect.

4.3 Combination of prosodic and non-prosodic feature types

Table 6 shows the most prominent outcomes measured in the matrix of feature combination experiments with prosodic and non-prosodic features. The upper section of the table illustrates the BoW vector combined with certain prosodic attributes. The middle section depicts the BoW vector combined with the dialogue history and certain prosodic attributes. The lower section lists results for sets where either the dialogue history is combined with a selection of prosodic and non-prosodic features, or all the non-prosodic features are combined with one or all prosodic features.

For the CTP task the baseline appears to be (slightly) beaten exclusively in those cases when the five before-previous system question types are used as fea-

	Current Turn / CTP		Previous Turn / PTP	
feature set	acc.	F	acc.	F
Baseline	65.2	62.4	82.9	75.3
Q*t* + BoW + All Prosodic	65.0±2.0	58.7±2.4	90.8±0.9	89.0±0.9
Q*t* + BoW + Ipause	65.8±2.2	61.2±2.8	91.0±1.1	89.1±1.1
Q*t* + BoW + Tempo	67.1±2.6	62.0±2.6	90.9±1.0	89.1±1.1
Q*t* + BoW + Duration	66.8±2.2	60.6±3.3	91.1±1.6	89.4±2.0
Q*t* + BoW + 5Q + Ipause	67.8±2.5	63.5±3.0	91.0±1.0	89.2±1.1
Q*t* + BoW + 5Q + Tempo	68.6±2.7	63.6±4.2	90.8±1.3	89.0±1.3
Q*t* + BoW + 5Q + Duration	69.5±3.0	64.3±4.3	91.1±1.3	89.4±1.5
Q*t* + All Nonpros + Dur	68.7±2.3	64.3±3.3	90.9±0.7	89.0±1.0
Q*t* + All features	68.9±2.5	64.4±4.4	90.6±0.9	88.7± 1.5
Q*t* + 5Q + BoW + All Pros	68.2±3.1	63.1±4.4	90.8±1.2	89.0±1.5
Q*t* + 5Q + All Pros	67.6±2.8	63.8±3.4	85.8±1.8	82.5±2.6
Q*t* + 5Q + BF + Dur	69.2±3.0	64.0±5.2	86.4±1.8	83.5±3.3

Table 6: Most prominent test performances in terms of accuracy and F-score trained on detecting miscommunication, based on feature type combination sets.

tures. These cases correspond to the set of Qt + 5Q and the other sets below it in Table 5, as well as to the sets in the two lower sections of Table 6. Taken alone or combined with any other feature, the F-score of these sets on the CTP task is around 64%. The best F-score of 65.0%, obtained by combining all non-prosodic features, is listed in Table 5, but all the other experiments on feature sets that include the six system questions (Qt + 5Q) perform non-significantly better or worse when tested in one-tailed *t*-tests. Likewise, combinations of prosodic and non-prosodic features for the PTP task are not significantly different from the apparent ceiling score of 91% accuracy and 89% F-score using non-prosodic features only (cf. the relevant feature sets in Table 5 and Table 6), provided that the combination includes the bag-of-words vector.

In sum, Table 6 confirms the findings of Tables 4 and 5 that –the prompt type condition imposed on the rule induction– (i) no (combination of) prosodic features plays an essential positive role in attaining top scores on either task; (ii) the combination of Q*t* and the five before-previous system question types is essential and sufficient for attaining an above-baseline score on the CTP task; and (iii) the combination of Q*t* and the bag-of-words vectors is essential and sufficient for reaching a ceiling score on the PTP task.

5 Discussion

In this paper we studied the usefulness of a wide range of features for machine-learning-based error detection in spoken dialogue systems. The features come from various sources, representing the dialogue history (the six most recent system question types), output of the ASR (recognized bag-of-words, acoustic con-

1. **if** Q (t) = R, **then** *Pr*. (977/0)
2. **if** Q (t) = I ∧ “naar” ∈ BoW(t-1)∧ “naar” ∈ BoW(t) ∧ “uur” ∉ BoW(t) **then** *Pr*. (131/12)
3. **if** Q (t) = I ∧ “naar’ ∈ BoW(t-1) ∧ “vanuit” ∈ BoW(t) **then** *Pr*. (39/6)
4. **if** Q(t) = I ∧ “uur” ∈ BoW(t-1) ∧ “om” ∈ BoW(t-1) **then** *Pr*. (44/7)
5. **if** Q(t) = I ∧ “van” ∈ BoW(t) ∧ “den” ∈ BoW(t) **then** *Pr*. (13/5)
6. **if** Q (t) = I ∧ “uur” ∉ BoW(t) ∧ “ik” ∈ BoW(t) ∧ “niet” ∈ BoW(t) **then** *Pr*. (11/2)
7. **if** Q (t) = E ∧ “nee” ∈ BoW(t) ∧ “ja” ∉ BoW(t)**then** *Pr*. (88/9)
8. **if** Q (t) = E ∧ “uur” ∈ BoW(t-1) ∧ “morgenochtend” ∈ BoW(t-1) **then** *Pr*. (10/2)
9. **if** Q (t) = E ∧ “uur” ∈ BoW(t) **then** *Pr*. (16/5)
10. **if** Q (t) = O ∧ “naar” ∈ BoW(t-1) **then** *Pr*. (38/9)
11. **if** Q (t) = O ∧ “wil’ ∈ BoW(t-1) **then** *Pr*. (6/1)
12. **if** Q (t) = O ∧ “naar” ∉ BoW(t) ∧ “februari” ∈ BoW(t) **then** *Pr*. (3/0)
13. **if** Q (t) = M ∧ “klopt” ∈ BoW(t-1) **then** *Pr*. (4/0)
14. **if** Q (t) = M ∧ “’s avonds” ∈ BoW(t-1) **then** *Pr*. (4/0)
15. **if** Q (t) = M ∧ “ik” ∈ BoW(t-1) **then** *Pr*. (8/2)
16. **if** Q (t) = Y ∧ “twee” ∈ BoW(t-1) ∧ “niet” ∈ BoW(t) **then** *Pr*. (2/0)
17. **else** ¬*Pr*. (2064/220)

Figure 2: The RIPPER rule set for the PTP task, on the basis of the most recent system question (Qt) plus the word graph of the current (BoWt) and the previous user input(BoW$t-1$). For translations of lexical items see the text. The (n/m) numbers at the end of each line indicate the number of instances the rule covers (n) and the number of false predictions (m).

fidence score, branching factor and amount of initial pause in the word graph) and various prosodic characteristics (pitch, loudness, tempo, duration). Two tasks were distinguished: predicting whether the current user utterance will cause communication problems (CTP) and identifying whether the previous user utterance caused communication problems (PTP). The CTP task is more difficult than the PTP task, since for predicting whether the previous user utterance caused problems the classifier can use the properties of the current user utterance, which may contain various cues indicating that something went wrong.

Concerning the CTP task, we see that none of the prosodic features yield above-baseline scores with our learning method. The best overall result for this task is obtained by training on all non-prosodic features (with an accuracy of 69.3% and an F-score of 65%). However, the improvement is only a few points above the baseline, and this implies that *a priori* prediction of problems is all but impossible for the current system. Earlier work, e.g., (Hirschberg, Litman, and Swerts, 2000; Hirschberg, Litman, and Swerts, 1999) has shown that prosody can help for both tasks, arguing that utterances which are produced with a marked intonation have a higher chance of being misrecognized and, moreover, if users speak with marked prosody this is also often an indication of problems in the previous turn. The descriptive statistics for our prosodic data do show that problematic and unproblematic utterances are significantly different from each other, but the learning algorithm fails to profit from these differences to the same extent as it does from other sources of information. A potential cause for not having been able to demonstrate the alleged added value of prosody is that our corpus consists of relatively short dialogues (2-10 turns) from more than 400 different speakers, whereas the corpus analysed by Hirschberg and co-workers consist of longer dialogues with 20 speakers. It might well be that prosody is more helpful when dialogues are longer, so that the system is better able to distinguish a person's regular speaking style from his/her problem-signalling speaking style.

The classifier does much better on the PTP task. We see that various prosodic features in isolation perform above baseline, and when training on all prosodic features we obtain 84.7% accuracy and 80.8% F-score. So it appears that using prosody is beneficial for this task. However, the benefits of prosody are relatively small when compared to those of other features. In particular, the combination of the dialogue history and the bags-of-words obtains 91.1% accuracy and 89.4% F-score. Combined with the only the most recent prompt type, the bag-of-words feature is able to capture situational patterns which otherwise had to be represented by high-level features. We applied RIPPER to the complete data in order to illustrate the rules learnt when training on the bag-of-words feature, cf. Figure 2.

Rule 2 captures situations when the user corrects the system in reply to its Implicit verification question: the lexical item "naar" ('to $prep.$') is present both in the current and in the previous word graph, whereas the lexical item "uur" ('o'clock') is not in the current word graph. Presumably the system made an implicit verification of the user's previous input on the arrival station, at the same time prompting for the time of travel, which the user is not giving in this situation, as s/he is concentrating on correcting the system. Rule 3 characterizes the user repeating his/her input in reply to an Implicit verification (of departure and arrival station) with a marked lexical usage: "vanuit" ('from') in the full form is present in the current word graph, whereas "naar" was present in the previous utterance's graph. Rule 6 points out problematic situations where the personal pronoun "ik" ('I') suggests that the user formulates the input with a full-fledged syntactic structure, a usage that is characteristic of problematic situations (Krahmer et al., 1999), as well as the parallel presence of the explicit disconfirmation marker "nee" ('no') and absence of the explicit confirmation marker "ja"('yes') in rule 7.

Rule 10 sheds light on problematic turn sequentions where, in reply to a Yes/No question 'Do you want information about another connection?' users often respond 'Yes, from X to Y', however, the system is unable to recognize the station names unsolicitedly given in the context of the Yes/No question –although it is clear from the rule that 'to $prep.$' ("naar") was in the word graph–, reacting with the usual Open question-prompt in the next turn: 'From where to where do you want to travel?'. Once in the Open question context, the ASR often fails to recognize the lexical item "naar", which is thus absent in the word graph (even though most probably the user did provide an answer for this slot), but traces of other, unsolicited information are present in the graph (for example the intended day of the travel, Rule 12).

We assume that the BoW set implicitly contains a wide spectrum of cues indicating problems, such as context shifts reflected by the differences between the current and the previous word graph, semantic diversity, syntactic structure, repetitions, omissions, corrections, and whether or not the system has recognized the necessary slot-filling item.

Our general result is that given the imposed-first-rule induction method, we can reach the level of the best results using varied sets of features, but these should at least include (1) the five before-previous prompt types for the CTP tasks, and (2) the two sets of bag-of-words for the PTP task. It is not possible to get a significantly higher F-score than obtained with these feature sets. On the other hand, performing no active selection but simply gather a large, assumed-to-be-comprehensive set of features did not produce significantly different results (cf. (Batliner et al., 1999) for a similar finding in a multi-feature prosody task). From a performance perspective, feature selection with RIPPER on the studied tasks has not been necessary. From an explanatory data analysis perspective, however, inspecting rules induced from selected features can pinpoint the most salient information that qualifies best to be related back to developers of SDS.

References

Batliner, A., E. Noeth, J. Buckow, R. Huber, V. Warnke, and H. Niemann. 1999. Prosodic feature evaluation: Brute force or well designed? In *Proccedings of the 14th Int. Congress of Phonetic Sciences*, volume 3, pages 2315–2318, San Francisco.

Batliner, A., E. Noeth, J. Buckow, R. Huber, V. Warnke, and H. Niemann. 2001. Duration features in prosodic classification: Why normalization comes second, and what they really encode. In *Proc. of ISCA Tutorial and Research Workshop*, pages 23–28, Red Bank, NJ.

Cohen, W. W. 1995. Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, Lake Tahoe, California.

Daelemans, W. and A. van den Bosch. 1992. Generalisation performance of backpropagation learning on a syllabification task. In M. F. J. Drossaers and A. Nijholt, editors, *Proc. of TWLT3: Connectionism and Natural Language Processing*, pages 27–37, Enschede. Twente University.

Greenberg, S. 2001. From here to utility? Melding phonetic insight with speech technology. In *Conference on Speech Communication and Technology (Eurospeech-2001)*, pages 2485–2488, Aalborg, Denmark.

Hermes, D. 1988. Measurement of pitch by subharmonic summation. *Journal of the Acoustical Society of America*, 83:257–264.

Hirose, K. 1995. Disambiguating recognition results by prosodic features. In Y. Sagisaka, N. Campbell, and N. Higuchi, editors, *Computing Prosody*. Springer, pages 327–342.

Hirschberg, J., D. Litman, and M. Swerts. 1999. Prosodic cues to recognition errors. In *Proceedings of the 1999 International Workshop on Automatic Speech Recognition and Understanding*, pages 349–352, Keystone, CO.

Hirschberg, J., D. Litman, and M. Swerts. 2000. Generalizing prosodic prediction of speech recognition errors. In *Proceedings of the 6th International Conference of Spoken Language Processing (ICSLP-2000)*, Beijing, China.

Krahmer, E., M. Swerts, M. Theune, and M. Weegels. 1999. Error spotting in human-machine interactions. In *Proceedings of the European Conference on Speech Communication and Technology, (EUROSPEECH '99)*, pages 1423–1426, Budapest, Hungary.

Krahmer, E., M. Swerts, M. Theune, and M. Weegels. 2001. The dual of denial: Two uses of disconfirmations in dialogue and their prosodic correlates. *Speech Communication*, 36(1):133–145.

Lendvai, P., A. Van den Bosch, E. Krahmer, and M. Swerts. 2002. Improving machine-learned detection of miscommunications in human-machine dialogues through informed data splitting. In *Proc. Workshop on Machine Learning Approaches in Computational Linguistics*. ESSLLI '02, Trento, Italy.

Litman, D., J. Hirschberg, and M. Swerts. 2001. Predicting user reactions to system errors. In *Proceedings of the 39th Annual Meeting and 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 362–369, Toulouse, France.

Litman, D. and S. Pan. 1999. Predicting and adapting to poor speech recognition in a spoken dialogue system. In *Proceedings of the 7th International Conference of User Modelling*, Banff, Canada.

Litman, D., M. Walker, and M. Kearns. 1999. Automatic detection of poor speech recognition at the dialogue level. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 309–316, New Brunswick, NJ. ACL.

Oviatt, S., M. McEachern, and G.-A. Levow. 1998. Predicting hyperarticulate speech during human-computer error resolution. *Speech Communication*, 24:87–110.

Salton, G. 1989. *Automatic text processing: The transformation, analysis, and retrieval of information by computer*. Addison–Wesley, Reading, MA, USA.

Shimojima, A., H. Koiso, M. Swerts, and Y. Katagiri. 1998. An informational analysis of echoic responses in dialogue. In *Proc. 20th Annual Conference of the Cognitive Science Society*, pages 951–956, Madison, WI, USA.

Swerts, M., D. Litman, and J. Hirschberg. 2000. Corrections in spoken dialogue systems. In *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP-2000)*, pages 615–618, Beijing, China.

Taylor, P., H. Shimodaira, S. Isard, S. King, and J. Kowtko. 1996. Using prosodic information to constrain language models for spoken dialogue. In *Proc. ICSLP '96*, volume 1, pages 216–219, Philadelphia, PA.

Van den Bosch, A., E. Krahmer, and M. Swerts. 2001. Detecting problematic turns in human-machine interactions: Rule-induction versus memory-based learning approaches. In *Proceedings of the 39th Meeting of the Association for Computational Linguistics*, pages 499–506, New Brunswick, NJ. ACL.

van Rijsbergen, C.J. 1979. *Information Retrieval*. Buttersworth, London.

Veilleux, N. M. and M. Ostendorf. 1994. Prosody/parse scoring and its application in ATIS. In *Proc. ARPA Human Language Technology Workshop '93*, pages 335–340, Princeton, NJ.

Walker, M., J. Wright, and I. Langkilde. 2000. Using natural language processing and discourse features to identify understanding errors in a spoken dialogue system. In *Proceedings of the International Conference on Machine Learning*, Stanford, CA.