Looking at the Last Two Turns, I'd Say This Dialogue is Doomed – Measuring Dialogue Success

Stefan Steidl[1], Christian Hacker[1], Christine Ruff[1], Anton Batliner[1], Elmar Nöth[1], and Jürgen Haas[2] *

[1] Lehrstuhl für Mustererkennung, Universität Erlangen-Nürnberg, Germany
steidl@informatik.uni-erlangen.de
[2] Sympalog Voice Solutions GmbH, 91052 Erlangen, Germany

Abstract. Two sets of linguistic features are developed: The first one to estimate if a single step in a dialogue between a human being and a machine is successful or not. The second set to classify dialogues as a whole. The features are based on Part-of-Speech-Labels (POS), word statistics and properties of turns and dialogues. Experiments were carried out on the SympaFly corpus, data from a real application in the flight booking domain. A single dialogue step could be classified with an accuracy of 83 % (class-wise averaged recognition rate). The recognition rate for whole dialogues was 85 %.

1 Introduction

Nowadays, the technology of automatic speech recognition and understanding systems for natural, spontaneous speech is that sophisticated that automatic speech dialogue systems are now able to support or, in some cases, even to replace traditional call centers. Numerous dialogue systems are working successfully in fields like car leasing, cinema ticket ordering, and miscellaneous information systems. The willingness of the companies to use these systems as well as the acceptance in the population is growing steadily. Still, switching to a new application domain means a huge effort to adapt the dialogue systems to the new scenario. Typically, this includes a change of the vocabulary and the language model used by the speech recognizer, and an adjustment of the dialogue manager. A practical and frequently used way is to develop a first system on the basis of no or only little user data from the new application scenario. The success rate of this first system will be very low. With the help of more and more user data from incoming calls, it is possible to improve the dialogue system step by step. Finally, this will result in a dialogue system with a high dialogue success rate. In real applications, one has to deal with a thousand and more calls per

* Parts of this work were funded by the European Commission (IST programme) in the framework of the PF-STAR project under Grant IST-2001-37599. The responsibility for the content lies with the authors.

day. To improve the dialogue system, especially those dialogues are of interest where something went wrong. Since it is laborious to work through all the data by hand, this inevitably brings up the question of how to find the abortive dialogues automatically. The work presented in this paper deals with exactly this question. On our way to measure the success of a whole dialogue, we classified single dialogue steps as successful resp. problematic. Using the success rates of the first n dialogue steps, the dialogue manager will be able to decide during the current dialogue, whether it makes sense to continue this dialogue or if it might be better to hand over to a human operator. This will be subject to future research. Future experiments will also show if knowledge about the emotional user states can help to classify the dialogue step success and, vice versa, if knowledge about the dialogue step success can help to classify user states.

2 The SympaFly Corpus

All our experiments were conducted on the SympaFly corpus which is described in this section. The SympaFly database was recorded using a fully automatic speech dialogue system for flight booking accessible via telephone and spontaneous speech. Following the approach pointed out in the introduction, the dialogue system was developed in three stages *S1*, *S2*, and *S3*. The dialogue success rates increased from about 30 % at the beginning to more than 90 % in the final system. Likewise, the average word error rate decreased from about 41 % in *S1* to less than 23 % in *S3*. The corpus consists of about eight hours of spontaneous German speech. There are 270 dialogues (137 of male and 133 of female speakers) available which comprise 6971 single dialogue steps (also called turns). It is possible that some users called the system several times. The callers had the task to book up to three flights. For more details see [1].

For all turns, the recognized word chain of the speech recognizer that was used during the corresponding development stage of the system as well as the actually spoken word chain of the calling user exists. In addition, several conversational peculiarities were annotated.[3] Firstly, different kinds of repetitions are labeled: exact repetitions where the information is repeated with exactly the same wording, semantic repetitions where the users utters the same information, but with different words, partial repetitions where only parts of the information (mostly the important ones) are repeated, or repetitions because of missing parts due to recording errors. All kinds of repetitions are subsumed under the label *REP*. The second group of labels (*BRK*) refers to breaks in the dialogue course: thematic breaks where the user's answer does not fit to the system's question, or meta speech where the user talks not to the system, but to someone else or to himself. The third conversational peculiarity are cases where the user does not answer at all (*NOA*). This often happens in situations where the user feels insecure.

[3] Further annotations exist for prosodic peculiarities on word level and emotional user states on turn level. For more information see [1].

Slot	Description
PlaceDeparture	place of departure
PlaceArrival	place of arrival
Date	date specification for arrival resp. departure
TypeDate	indicates if it's a date of arrival or departure
Time	time specification for arrival resp. departure
TypeTime	indicates if it's a time of arrival or departure
NumPersons	number of requested tickets
Class	class: first class, business class or economy class
QualiflyerYN	indicates if the user is a member of the frequent flyer program
QualiflyerNo.	frequent flyer number
CreditCardNo.	credit card number
ExpiryDate	expiry date of the credit card
YesNo	indicates if "yes" resp. "no" or equivalent statements are uttered
DMarker	indicates a jump to another dialogue section

Table 1. Slots that are automatically filled by a parser

The purpose of our work is to classify dialogues as successful or abortive. In order to evaluate the classifier, a reference is needed which was also annotated. Dialogues are labeled as successful if the system was able to book at least the outward flight. In the SympaFly corpus, 161 dialogues are categorized as successful, 109 as problematic. As mentioned above, we also classified single dialogue steps as successful or problematic. The reference for the dialogue steps was calculated automatically. We used the same mechanism as the speech understanding component. A parser, based on semantic units, extracted the relevant information out of the recognized word chain of the speech recognizer. The results are slots that are filled with the corresponding information. Table 1 lists the names and the meanings of the most important slots used in the SympaFly system. We now let the parser fill the slots on the recognized word chain as well as on the actually spoken word chain and compared both results. A single dialogue step was considered to be successful if the same slots were filled and the information was the same in all slots. Otherwise this turn was considered to be problematic. This procedure assumes that the parser works perfectly and is a very rigid criterion.

3 Linguistic Features

In this section, the features used to classify the dialogue step success or the dialogue success are described. We focused on linguistic, not on acoustic features. Our features are based on the uttered word sequence, either the actually spoken words or the word chain recognized by the speech recognizer. The acoustic signal is not needed. In other studies focusing on trouble in human–machine communication, indicators are, for instance, recognition errors [2], user corrections [3], the user hangs up, a wizard has to take over, or a task fails completely [6].

3.1 Dialogue Step Success Features

We have two groups of features for classifying the dialogue step success. The features of the first group are based on the actually spoken word chain and on annotations. We call these features *unfair*, because neither the actually spoken word chain nor the annotations will be available in a running system. Nevertheless, they are useful as they are an upper limit if they are compared with their counterparts of the second group of *fair* features which are based on the recognized word chain of the system and on automatically filled slots.The group of *unfair* features:

- Repetitions of slots on the actually spoken word chain (*RepSlots_spoken_rel*, *RepSlots_spoken_abs*): Relative frequency and absolute number of the filled slots that are repeated in the next dialogue step with identical content. Repetitions should be a clear hint that the dialogue system did not understand the user right. The slot parser works on the actually spoken word chain.
- Conversational annotations (*NOA*, *BRK*, *REP*, *CONV*): *NOA*, *BRK*, and *REP* are set to 1 if the corresponding conversational label is annotated in the next dialogue step. *CONV* equals 1 if at least one of these three labels is annotated in the following turn.
- The feature *Yes_spoken* is set to 1 if the parser sets the slot *YesNo* to "yes" in the current dialogue step. "Yes" or equivalent statements are often uttered if the user wants to confirm a system's question.

The group of *fair* features:

- Average a posteriori word probability (*Confidence*): This confidence measure is not a linguistic but an acoustic feature which states how sure the speech recognizer is that a word in its output word chain is correct (see [5]). This feature is included as it is part of the speech recognizer's output.
- Repetitions of slots on the recognized word chain (*RepSlots_recog_rel*): Similar to the unfair feature *RepSlots_spoken_rel*, but calculated on the recognized word chain. Furthermore, slots are considered to be repeated if they are filled again in the next dialogue step independent if the content of the slot is the same or not. This is necessary due to recognition errors.
- Length of a dialogue step (*Turnlength*, *Turnlength_theta*): *Turnlength* is the number of words in a dialogue step. *Turnlength_theta* is set to 1 if the number of words is less or equal to a given threshold θ. Best results were achieved with a threshold $\theta = 4$. In general, shorter turns were more successful than problematic ones in our system. Successful turns had an average length of 3.0 words, while problematic turns consisted of 6.2 words on average.
- POS classes (*Unigr_NOUN*, *Unigr_APN*, *Bigr_APN_APN*, *Bigr_PAJ_NOUN*, *Bigr_NOUN_PAJ*): According to preliminary investigations, we used promising uni- and bigrams. The features count how often the corresponding Part–of–speech category resp. pair of POS categories occurs in a dialogue step. All POS features are normalized with the turn length. The category *NOUN*

represents nouns, proper names, single letters, and fragments of nouns. Participles and adjectives in their basic form belong to the category *APN*, articles, particles, and interjections to the category *PAJ*. Other POS categories were not used as features.

- Touch tones (*Touch_tones*, *Touch_tones_next*): In the SympaFly system, the user enters his credit card number via the telephone keys (dual tone multi frequency, DTMF). These features indicate whether this milestone is reached in the current resp. next dialogue step.
- Turn number (*TurnNo.*): All dialogue steps are numbered chronologically from the beginning of the dialogue. In the SympaFly corpus, successful dialogues consist of 33.2 turns on average, while abortive ones are only 15.9 turns long. Hence, a higher turn number is an indicator for a successful dialogue step. This may be a peculiarity of our scenario where the user's task was to book up to three flights: If the first booking is successful, the user will probable be able to book the second or third one, too. In contrast, the user often hangs up, if he fails to book the first flight.
- Dialogue events (*DMarker*): The course of the dialogue is determined by certain utterances of the user. If the user says goodbye, the parser sets the slot *DMarker* to "exit", for example. Other possible values are "menu", "help", "restart" etc. This feature is set to 1 if the slot *DMarker* is set to "exit" in the following dialogue step since this is considered to be positive. It is set to -1 for all other values of *DMarker* and it it set to 0 if the slot *DMarker* is not set at all in the next turn.
- Yes/No-Features (*Yes*, *Yes_next*, *Yes_last*, *No_next*, *No_last*, *No_length*, *Yes_confidence*, *YesNo*): The features *Yes*, *Yes_next* and *Yes_last* indicate whether the slot *YesNo* is set to "yes" in the current, the next resp. the last dialogue step. Accordingly, *No_next* and *No_last* indicate if the slot *YesNo* is set to "no" in the corresponding dialogue step. The feature *Yes* is the counterpart of the *unfair* feature *Yes_spoken*. Often, "no" in the user's answer is a sign that the system did not understand the user right. In many cases, he then uses a long correction turn like "No, I want to fly from Berlin to New York". In contrast, in short turns, "no" can also be the answer to a question like "Are you a member of the frequent flyer program?". Hence, in these cases, "no" is not a sign for problematic dialogue steps. The feature *No_length* tries to compensate for this. It is set to the length of the next turn, if the slot *YesNo* is set to "no" in the following step. Because of frequent recognition errors of the word "yes", we introduced the feature *Yes_confidence*, a combination of the feature *Yes* and our confidence measure for the word "yes". *YesNo_next* is a combination of *Yes_next* and *No_next*: It is 1 if the slot *YesNo* is set to "yes" in the next turn, -1 if *YesNo* is set to "no", and 0 otherwise.
- Filled Pauses (*FilledPauses*, *FilledPauses_next*): Filled pauses like "uh" or "uhm" are a sign of hesitation or unsureness. These two features indicate if filled pauses occur in the current resp. next turn.

3.2 Dialogue Success Features

Motivated by the results of the fair dialogue step success features (results will be presented in Section 4.1), we only implemented fair features to classify the success of the whole dialogue:

- Portion of the successful resp. problematic dialogue steps in a dialogue (*Portion_100*, *Portion_0*): A single dialogue step failure does not indicate that the whole dialogue is abortive, of course. But the likelihood that a dialogue fails increases with the relative frequency of problematic dialogue steps.
- Length of the longest chain of successful resp. problematic dialogue steps (*LongestSequence_100*, *LongestSequence_0*): After a long sequence of problematic turns, the probability is high that the caller hangs up.
- Average dialogue step success (*AvgDSS*): For each single step in a dialogue, the dialogue step success is calculated and then averaged over all steps.
- Average dialogue step success of the last two resp. three steps in a dialogue (*AvgDSS_last2*, *AvgDSS_last3*)
- Number of slots which were filled during a dialogue (*NumFilledSlots*): One condition precedent for a successful flight booking is that the system was able to gather all the necessary information, that means to fill all the slots.
- Absolute number of dialogue steps where the slot *YesNo* is set to "no" resp. "yes" (*No_abs*, *Yes_abs*)
- Relative frequency of the dialogue steps where the slot *YesNo* is set to "no" resp. "yes" (*No_rel*, *Yes_rel*)
- Number of dialogue steps in a dialogue (*NumTurns*): Longer dialogues are more likely to be successful than shorter ones. See feature *TurnNo.*
- Number of repeated slots in a dialogue (*Rep_abs*): A high number of repetitions is a sign that the system performs not very well.
- Number of repeated slots normalized with the length of the dialogue (*Rep_abs_length*)
- User thanks the system at the end (*Thanks*): The German word "danke" (thank you) is uttered in the last dialogue step.
- Dialogue ends with a farewell (*Exit*): The slot *DMarker* is set to "exit" in the last dialogue step.

4 Experiments

4.1 Classifying Dialogue Step Success

To evaluate the benefit of each dialogue step feature, we classified each feature on its own. The results in Table 2 were achieved using decision trees from the Edinburgh Speech Tools Library. Train and test set were disjoint. The recognition rates are calculated per class and then averaged over both classes. For the final classifier, we abandoned our unfair features. As our database is rather small, we reduced the number of features to ten. By taking those features that were closest to the root of a decision tree trained with all features, we obtained

Feature	CR [%]	Feature	CR [%]	Feature	CR [%]
RepSlots_spoken_rel	*56.2*	RepSlots_recog_rel	55.3	Yes	67.2
RepSlots_spoken_abs	*56.2*	Unigr_NOUN	61.9	Yes_next	50.0
NOA	*50.0*	Unigr_APN	65.6	Yes_last	50.0
BRK	*50.9*	Bigr_APN_APN	54.3	No_next	54.3
REP	*53.3*	Bigr_PAJ_NOUN	61.1	No_last	50.0
CONV	*50.0*	Bigr_NOUN_PAJ	56.7	No_length	51.9
Yes_spoken	*68.4*	Touch_tones	50.0	Yes_confidence	68.5
Confidence	69.2	Touch_tones_next	50.0	YesNo_next	54.3
Turnlength	65.2	TurnNo.	51.4	FilledPauses	50.0
Turnlength_theta	65.2	DMarker	51.9	FilledPauses_next	50.0

Table 2. Separate classification of all dialogue step success features (class-wise averaged recognition rates, CR). Unfair features are printed in italics

the following features: *Confidence*, *Turnlength*, *TurnNo.*, *Touch_tones*, *NOUN*, *APN*, *Bigr_PAJ_NOUN*, *YesNo_next*, *Yes_confidence*, and *No_length*. Using Support Vector Machines (LIBSVM) and cross validation we achieved a class-wise averaged recognition rate of 82.5 %.

4.2 Classifying Dialogue Success

At first, we also evaluated the dialogue success features on their own. Because of the small size of the corpus (270 dialogues), we used a Leave–One–Out strategy. Class-wise averaged recognition rates obtained with decision trees are given in Table 3. Again, we reduced the number of features and used only those six features which were closest to the root of a decision tree trained with all features. The best features were: *Yes_abs*, *No_abs*, *LongestSequence_0*, *AvgDSS_last2*, *Rep_abs*, and *NumTurns*. The best result, which we also obtained with decision trees, was a class-wise averaged recognition rate of 85.4 %. The Receiver Operating Characteristic (ROC) curve, which was produced with Support Vector Machines, is given in the left part of Fig. 1. The true positive rate, also called sensitivity, is the number of abortive dialogues that were classified correctly as abortive. The false positive rate (1 − specificity) is the number of successful dialogues that were classified wrongly as abortive. Choosing a specificity of 91.9 % and a sensitivity of 74.3 %, the error analysis has to be done on only 94 out of 270 dialogues (34.8 %, Fig. 1 right). Only 13 of these dialogues are classified wrongly, 81 out of 109 abortive dialogues are captured.

5 Conclusion and Outlook

Our experiments show that we can predict with simple linguistic features if a single dialogue step or a whole dialogue is successful or not. When switching to a new application scenario, this can be very helpful to select only those dialogues where something went wrong as those abortive dialogues are needed to improve

Feature	CR [%]	Feature	CR [%]	Feature	CR [%]
Portion_0	65.3	AvgDSS_last3	70.6	NumTurns	67.3
Portion_100	63.7	NumFilledSlots	61.4	Rep_abs	63.5
LongestSequence_0	50.0	No_abs	58.2	Rep_abs_length	63.5
LongestSequence_100	65.4	No_rel	63.2	Thanks	50.0
AvgDSS	65.7	Yes_abs	80.8	Exit	50.0
AvgDSS_last2	66.4	Yes_rel	63.4		

Table 3. Separate classification of all dialogue features

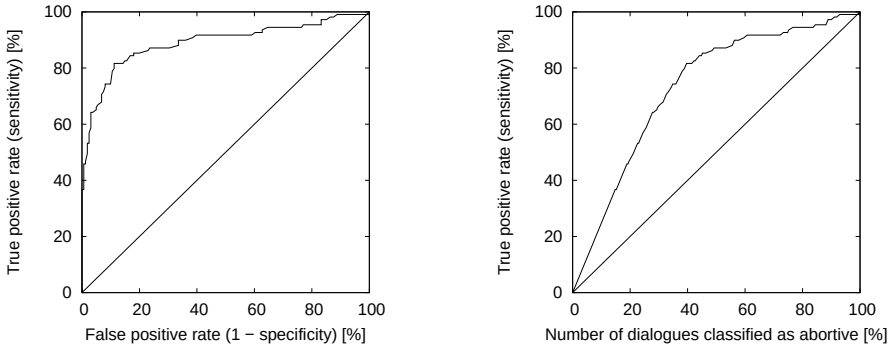

Fig. 1. Classification of the dialogue success: ROC curve (left) and number of dialogues classified as abortive vs. true positive rate (right)

the dialogue system. Future research will investigate whether the dialogue step success classification of the first n turns can be used online to decide if in case of trouble, the dialogue should be continued by the system or if it might be better to hand over to a human operator. We will also investigate if the dialogue step success can help to classify emotional user states and vice versa, if knowledge about emotional user states can improve the classification of dialogue steps.

References

1. A. Batliner, C. Hacker, S. Steidl, E. Nöth, and J. Haas: User States, User Strategies, and System Performance: How to Match the One with the Other. In: Proc. of ISCA-EHSD '03, p. 5-10, Château d'Oex, 2003
2. J. Hirschberg, D. Litman, and M. Swerts: Prosodic Cues to Recognition Errors. In: Proc. of ASRU '99, pages 349–352, Keystone, 1999
3. G.-A. Levow: Characterizing and Recognizing Spoken Corrections in Human-Computer Dialogue. In: Proc. of COLING/ACL '98, p. 736-742, Montréal, 1998
4. C. Ruff: Bestimmung des Dialog(schritt)erfolgs mit linguistischen Merkmalen. Studienarbeit, Lehrstuhl für Mustererkennung, Universität Erlangen-Nürnberg, 2004
5. G. Stemmer, S. Steidl, E. Nöth, H. Niemann, and A. Batliner: Comparison and Combination of Confidence Measures. In: Proc. of TSD '02 , p. 181-188, Brno, 2002
6. M. A. Walker, I. Langkilde J. Wright, A. Gorin, and D. Litman: Learning to Predict Problematic Situations in a Spoken Dialogue System: Experiments with How May I Help You? In: Proc. of NAACL '00, p. 210-217, Seattle, 2000