ISCA Archive
http://www.isca-speech.org/archive

6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

PROBLEM SPOTTING IN HUMAN-MACHINE INTERACTION

Emiel Krahmer, Marc Swerts, Mariët Theune, Mieke Weegels

IPO, Center for Research on User-System Interaction, Eindhoven, The Netherlands
{E.J.Krahmer/M.G.J.Swerts/M.Theune/M.F.Weegels}@tue.nl

10.21437/Eurospeech.1999-322

ABSTRACT

In human-human communication, dialogue participants are continuously sending and receiving signals on the status of the information being exchanged. We claim that if spoken dialogue systems were able to detect such cues and change their strategy accordingly, the interaction between user and system would improve. Therefore, the goals of the present study are as follows: (*i*) to find out which positive and negative cues people actually use in human-machine interaction in response to explicit and implicit verification questions and (*ii*) to see which (combinations of) cues have the best predictive potential for spotting the presence or absence of problems. It was found that subjects systematically use negative/marked cues (more words, marked word order, more repetitions and corrections, less new information etc.) when there are communication problems. Using precision and recall matrices it was found that various combinations of cues are accurate problem spotters. This kind of information may turn out to be highly relevant for spoken dialogue systems, e.g., by providing quantitative criteria for changing the dialogue strategy or speech recognition engine.

1. INTRODUCTION

A major issue in present day spoken dialogue system design is how to deal with communication problems. Since a spoken dialogue system can never be certain that it understood the user correctly, it is in a constant need for verification of its current assumptions. After a question like (1) of a user (U) who interacts with a train time table information system (S), the system could check that it understood the user's utterance correctly by using either an *explicit* or an *implicit* verification strategy ((2.a) and (2.b) respectively).

(1)	U	I want to go to Swalmen.
(2.a)	S	Do you want to go to Swalmen? (*explicit*)
(2.b)	S	When do you want to travel to Swalmen? (*implicit*)

The utterance in (2.a) is solely aimed at verifying that the system's current assumptions are correct. The main disadvantage of explicit verification is that it requires extra turns, which users may find annoying. Therefore, many systems opt for an alternative, *implicit* verification strategy. An example is (2.b). Here the system utterance has a double intention: it asks the user for a new piece of information (the desired moment of travelling) and at the same time it verifies the arrival station. The advantage of this strategy is obvious: when the system is not mistaken in its assumptions, this strategy is efficient and at the same time the resulting dialogue is much more fluent. The downside, however, is that when the system's assumptions are incorrect, users become confused (see e.g., [7]). For instance, if the user wants to correct the arrival station which (2.b) implicitly tries to verify, more effort is required since the user has to indicate that he or she will not answer the question asked (travel time) but rather react to one of its underlying assumptions. In sum: neither explicit nor implicit verification is by itself a satisfactory solution for dealing with the uncertainties in human-machine conversation.

From human-human conversation it is known that dialogue participants are continuously sending and receiving signals on the status of the information being exchanged. This process of *information grounding* ([2, 6]) typically proceeds in two phases: a *presentation* phase in which the current speaker sends a message to his conversation partner, and an *acceptation* phase in which the other signals whether the message came across correctly or not. The signals in the acceptation phase can either be positive ('go on') or negative ('go back'). It seems a valid assumption that the negative cues are comparatively marked, as if the speaker wants to devote additional effort to make the other aware of the apparent communication problem ([5]). This is most likely due to the fact that missing a negative cue has relatively more serious consequences than missing positive feedback: it may cause breakdown of the communication.

There are, to the best of our knowledge, not many dialogue systems that systematically keep track of the whole gamut of positive and negative cues that a user may send. We conjecture that when systems are able to immediately detect such cues and change their dialogue strategy accordingly, the fluency of the interaction will be improved. The goal of the present study is therefore twofold: (*i*) to find out which positive and negative cues people actually use in human-machine interaction in response to explicit and implicit verification questions (section 3.1) and (*ii*) to see which (combinations of) cues have the best predictive potential for spotting the presence or absence of problems (section 3.2). In section 4, a number of ways in which dialogue systems can change their strategy based on cue-detection are discussed. First, the method is described.

2. METHOD

For the analysis, a corpus (see [7]) was used consisting of 120 dialogues with two speaker-independent Dutch spoken dialogue systems which provide train time table information. The systems prompt the user for unknown slots, such as departure station, arrival station, date, etc., in a series of questions. The two systems dif-

fer mainly in verification strategy (one primarily uses implicit verification, the other only uses explicit verification), length of system utterances and speech output (concatenated vs. synthetic speech). Twenty subjects were asked to query both systems via telephone on a number of train journeys. They were asked to perform three simple travel queries on each system (in total six tasks). Two similar sets of three queries were constructed, to prevent literal copying of subjects' utterances from the first to the second system. The order of presenting systems and sets was counterbalanced.

From the 120 dialogues, all implicit and explicit verification questions and user's reactions to these were selected, giving 487 utterance pairs. A set of 44 pairs (proportionally distributed over the subjects) was not used for the analysis. This set consisted of three classes: (*i*) cases in which the user either accidentally or on purpose accepted a wrong result, (*ii*) cases in which the user was interrupted and thus could not properly "accept" the verification contribution initiated by the system and (*iii*) a limited number of cases in which subjects started their own "contribution" (e.g., ask a non-related question such as "Can I use my reduction card?"). The distribution of the 443 resulting adjacency pairs is given in Table 1. A communication problem arises iff the information which the system attempts to verify results from a speech recognition error (substitution, insertion or deletion) or is based on an incorrect default assumption (e.g., the system assumes that the user wants to travel *today*).

Table 1: Numbers of adjacency pairs containing no communication problems (¬ PROBLEMS) and those containing one or more problems (PROBLEMS), as a function of verification strategy.

	¬ PROBLEMS	PROBLEMS	TOTAL
EXPLICIT	211	116	327
IMPLICIT	87	29	116
TOTAL	298	145	443

The data were labeled as follows. Features of system utterances that were labeled include the number of verified slots, the presence of default assumptions and the number, type and recurrency of recognition errors. Of user utterances the following features were labeled: whether or not the user gave an answer, utterance length (number of words), word order, (dis)confirmation, and amount of repeated, new or corrected slots. For each feature a positive and a negative variant was operationalized. Based on the Principle of Minimal Collaborative Effort ([1]), it was assumed that both user and system want the dialogue to be finished successfully as soon as possible, and that they do not want to spend more effort than necessary for current purposes. If the preceding verification question contains a problem, users are expected to spend more effort on their signals in order to prevent complete breakdown of the communication. This leads to the distinction between positive and negative cues in table 2.

Table 2: Positive versus negative cues.

POSITIVE ('go on')	NEGATIVE ('go back')
short turns	long turns
unmarked word order	marked word order
confirm	disconfirm
answer	no answer
no corrections	corrections
no repetitions	repetitions
new info	no new info

The positive cues can be seen as unmarked settings of the features. For instance, the expected answer to a verification question is a confirmation, and the default word order in a sentence is unmarked (thus, no topicalization or extraposition). Additionally, it follows from the Principle of Minimal Collaborative Effort that it is a positive signal to present new information (which may speed up the dialogue), but not to repeat or correct information (which will definitely not lead to a more swift conclusion of the conversation). The central hypothesis can now be stated as follows: users more often employ the 'go back' signals when the preceding system utterance contains a problem, whereas the 'go on' signals are used in response to unproblematic system utterances. Additionally, it is expected that a 'go back' signal following an implicit verification will contain relatively more marked features than a 'go back' signal following an explicit verification.

3. RESULTS

3.1. Distribution of positive and negative cues

For all cues, it was found that there is no significant difference in user's reactions to recognition errors or to incorrect default assumptions. Therefore no distinction is made between these two sources of communication problems in analysing the data. Table 3 lists the average length in words of the user utterances. It confirms our central hypothesis: subjects use more words when there are problems, irrespective of verification strategy, where the average number of words is the highest in response to a problematic implicit verification question.

Table 3: Average number of words in user's utterances (standard deviations are given between brackets).

	¬ PROBLEMS	PROBLEMS
EXPLICIT	1.68 (1.68)	3.44 (3.19)
IMPLICIT	3.21 (2.09)	7.12 (2.10)

Table 4 contains the percentages of empty turns in the four cases of interest. These figures are comparatively low, due to the fact that empty turns were not often encountered ($n = 9$). Still it is interesting to point out that the distribution of empty turns follows the predicted trend: they arise more often when there is a problem, in particular following an implicit verification.

Table 4: Percentages of empty turns.

	¬ PROBLEMS	PROBLEMS
EXPLICIT	0%	2.6%
IMPLICIT	3.4%	10.3%

Table 5 records the relative frequency of turns with a marked word order (topicalization or extraposition). Again: the percentage of user utterances containing a marked word order is higher when there are communication problems, albeit that the difference is relatively small in the case of explicit verifications. Additionally, it is found that implicit verifications containing a problem are associated with the highest percentage of marked word orders by far.

Table 5: Percentages of turns with a marked word order.

	¬ PROBLEMS	PROBLEMS
EXPLICIT	3.3%	4.4%
IMPLICIT	1.2%	26.9%

Table 6 shows the respective percentages of (dis)confirmations, again as a function of the verification strategy. It is found that for explicit verifications the number of non-confirmations (not-"yes" answers) increases when there are problems. Similarly, for implicit verification, it turns out that the percentage of turns containing an explicit disconfirmation ("no") increases when there are problems: 15.4 % of the user's utterances contains a "no", even though the implicit verification question is not a yes/no-question. It is impossible to determine whether the 84.6 % other answers to problematic implicit verifications should count as disconfirmations. Hence, it is not possible to compare the amount of negative cues across verification strategies.

Table 6: Percentages of "yes" (right, sure …), "no" and other answers.

		¬ PROBLEMS	PROBLEMS
EXPLICIT	yes	92.8%	6.1%
	no	0%	56.6%
	other	7.1%	37.1%
IMPLICIT	yes	0%	0%
	no	0%	15.4%
	other	100%	84.6%

The final group of cues to be discussed is concerned with information units, measured in terms of slots. Table 7 illustrates that subjects repeat and correct more information when there are communication problems, and that they both repeat and correct most following problematic implicit verifications. Explicit verifications only occasionally lead users to provide new information, more or less independent of the presence of problems. It is interesting to note that for implicit verifications, on the other hand, the percentage of turns containing new information drastically decreases in the case of problems.

Table 7: Percentages of turns with repeated, corrected or new slots.

		¬ PROBLEMS	PROBLEMS
EXPLICIT	repeated	8.5%	23.9%
	corrected	0%	72.6%
	new	11.4%	12.4%
IMPLICIT	repeated	2.4%	61%
	corrected	0%	92.3%
	new	53.6%	36.5%

In conclusion: much support is found for the general hypothesis stated above. In nearly all cases subjects use negative cues more often when there are communication problems. Additionally, negative cues are used most often following an implicit verification containing a problem.

3.2. Problem spotting

The next question is: which cues provide useful information for a spoken dialogue system in determining that the communication is in trouble? To determine this, precision and recall matrices can be used. Consider the following contingency table for spotting communication problems.

	PROBLEMS	¬ PROBLEMS
PROBLEM SPOTTED	a	b
¬ PROBLEM SPOTTED	c	d

Elements of class a (there is a problem and this is signalled) are called *hits*. Elements of classes b, c and d are referred to as *false alarms*, *misses* and *correct rejections* respectively. Both precision and recall are defined in terms of this contingency table: *precision* $= \frac{a}{a+b}$, while *recall* $= \frac{a}{a+c}$. A high precision entails few false alarms, while a high recall corresponds with a low number of misses. From the current perspective, *precision* can be interpreted as follows: given that the user utters a certain negative cue, how certain can the system be that it is a reaction to a problem. *Recall* can be understood the other way round: given that a verification message from the system contains a problem, what are the chances that the following user utterance contains a certain negative cue. Obviously, a dialogue system would be perfectly able to spot problems if it has full precision and total recall.

Table 8: Precision and recall percentages for negative cues (single conditions), both for explicit and implicit verification.

		EXPLICIT		IMPLICIT	
	CONDITION	precision	recall	precision	recall
a	nr. words > 8	73%	10%	86%	23%
b	disconfirm.	100%	57%	100%	18%
c	no confirm.	94%	88%	24%	100%
d	marked w.o.	42%	4%	88%	27%
e	no answer	100%	3%	50%	10%
f	rep. slots > 0	60%	24%	89%	62%
g	corr. slots > 0	100%	73%	100%	92%
h	new slots = 0	35%	88%	24%	62%

Table 8 contains the precision and recall results for the negative cues discussed in the previous section. In the case of scalar cues (such as length of user utterance) only the optimal condition is listed (in this case, number of words is greater than 8). Unsurprisingly, following an explicit verification the single best cue for spotting errors is the absence of a confirmation (*c*), with 94 % recall and 88% precision, while following an implicit verification the overall most informative cue is a non-zero number of corrections (*g*), yielding a 100% precision and a 92% recall. Interestingly, following an explicit verification, a disconfirmation ("no") has a significantly lower recall than a non-confirmation (not "yes"). Note also that, following an implicit verification, the conditions *a* (nr. words > 8), *b* (disconfirmation) and *d* (marked word order) all have a high precision (thus: are good cues for spotting errors); unfortunately they also have a relatively low recall (due to their infrequency). An interesting question therefore is whether combinations of cues can overcome these limitations. Table 9 contains a number of such combinations.

Table 9: Precision and recall percentages for negative cues (boolean combination), both for explicit and implicit verification. The single conditions *a* through *h* refer to those in table 8.

	EXPLICIT		IMPLICIT	
CONDITION	precision	recall	precision	recall
$a \vee b \vee d$	88%	64%	83%	38%
$c \vee g$	88%	94%	24%	100%
$a \vee f \vee g$	82%	95%	89%	92%
$a \vee d \vee g$	91%	95%	92%	92%
$a \vee b \vee d \vee e$	88%	65%	72%	45%
$a \vee c \vee d \vee e$	82%	95%	25%	100%
$f \vee g$	82%	73%	92%	92%
$(f \vee g) \wedge h$	92%	61%	100%	58%
$f \wedge g$	100%	24%	100%	62%

For implicit verification, the disjunctive condition $a \vee b \vee d$ still has a relatively high precision and certainly a higher recall than either *a*, *b* or *d* in isolation, but still nothing to write home about. For explicit verification, both precision and recall of the disjunction $a \vee b \vee d$ are much higher than those of either disjunct in isolation. What is interesting about $a \vee b \vee d$ is that it only consists of cues which are concerned with the *form* of the user's utterance. However, the overall best condition is a mixture of 'form' and 'content': $a \vee d \vee g$ (i.e., user's reaction has more than 8 words or uses a marked word order or contains corrected information). It not only has a high precision (though not as high as *g* itself), but also a high recall.

Of course, paying attention to 'go back' signals is only one side of the coin. For many applications it is also of interest to keep track of the 'go on' signals. The question then is: which cue(s) provide useful information in determining that the communication is running smoothly? Table 10 contains the precision and recall results for both explicit and implicit verification for single conditions derived from the positive cues discussed in section 3.1.

Table 10: Precision and recall percentages for positive cues (single conditions), both for explicit and implicit verification.

		EXPLICIT		IMPLICIT	
	CONDITION	precision	recall	precision	recall
a	nr. words < 6	69%	97%	94%	87%
b	confirm.	97%	93%	0%	0%
c	no disconf.	81%	100%	79%	100%
d	unmark w.o.	65%	97%	81%	99 %
e	answer	65%	100%	76%	97 %
f	rep. slots $= 0$	69%	91%	89%	98 %
g	corr. slots $= 0$	87%	100%	98%	100 %
h	new slots > 0	63%	11%	82%	54 %

What is remarkable about table 10 is that the recall results for nearly all conditions are very high and often much higher than for their negative counterparts in table 8. Additionally, there is one condition for each validation strategy with a very high precision as well: this is *b* (confirmation) for explicit verification and *g* (no corrections) for implicit verification. Boolean combinations of these positive cues have also been studied, but none of these improved upon the scores for *b* or *g* and due to lack of space these combinations are not further discussed here.

4. DISCUSSION

Summarizing the main results: for nearly all cues studied it was found that subjects use the negative variants ('go back') more often when the preceding system utterance contains a problem, whereas the positive cues ('go on') are more often used in response to unproblematic system utterances. Additionally, a 'go back' signal following an implicit verification almost always contains more marked features than a 'go back' signal following an explicit verification. These findings provide potentially useful information for spoken dialogue systems which monitor whether the communication is in trouble or not: if a verification question is followed by a user's utterance which is relatively long, contains a marked word order or corrected information, the system can be fairly certain that the information it tried to verify is not in agreement with the user's intentions. If, on the other hand, the user's utterance contains a confirmation (in reaction to an explicit verification) or no correction (after an implicit verification), then it is highly likely that the verified information is correct. This kind of information can be very useful in a number of situations. For instance, in section 1 it was noted that neither implicit nor explicit verification is by itself a satisfactory solution for dealing with the uncertainties in human-machine dialogue. An attractive compromise would be to use implicit verification when the user sends 'go on' signals and switch to explicit verification when the user sends (continued) 'go back' signals. Another situation in which it might pay off to look at positive and negative cues is the following. Levow [4] found that the probability of experiencing a recognition error after a correct recognition is 16%, but immediately after an incorrect recognition it is 44%. This increase is probably caused by the fact that speakers use hyperarticulate speech when they notice that the system had a problem recognizing their previous utterance. It would be interesting to see whether the recognition results improve when the system switches to an speech recognition engine trained on hyperarticulate speech after a problematic system utterance and back again to the 'standard' recognizer when the communication is on the right track again.

Besides such applications, there are various other lines for future research worthy of exploration. First, notice that only combinations of a limited number of cues were studied. One may suspect that complex boolean combinations may yield even higher precision and recall results. It would also be interesting to repeat the experiments on dialogues obtained with other systems, to see to what extent the present findings are tied to the particulars of the systems used for collecting the corpus. Finally, notice that in this article prosody was not discussed at all. In [3], we make up for this lack. There it is shown that one of the central hypotheses of this paper carries over to prosody: in the case of communication problems, speakers put much more *prosodic* effort in their reaction.

Acknowledgments The authors are mentioned in alphabetical order. Weegels and Theune were supported by the Priority Programme Language and Speech Technology (TST), sponsored by NWO (The Netherlands Organization for Scientific Research). Swerts is also affiliated with UIA and with the FWO - Flanders. Krahmer was partly supported by the project LE-1 2277 (VODIS).

5. REFERENCES

1. Clark, H.H. & D. Wilkes-Gibbs (1986), Referring as a Collaborative Process, *Cognition* 22:1-39.
2. Clark, H.H. & E.F. Schaeffer (1989), Contributing to Discourse, *Cognitive Science*, 13:259-294.
3. Krahmer, E., M. Swerts, M. Theune & M. Weegels (1999), Prosodic Correlates of Disconfirmations, in: *Proc. ETRW on Dialogue and Prosody*, Eindhoven, The Netherlands.
4. Levow, G.A. (1998), Characterizing and Recognizing Spoken Corrections in Human-Computer Dialogue, in: *Proc. COLING-ACL*, Montreal, Canada.
5. Swerts M., H. Koiso, A. Shimojima & Y. Katagiri (1998), On different functions of repetitive utterances. in: *Proc. ICSLP-98*, Sydney, Australia.
6. Traum, D.R. (1994), *A Computational Theory of Grounding in Natural Language Conversation*, Ph.D thesis, Rochester.
7. Weegels, M. (1999), Users' (Mis)conceptions of a Voice-Operated Train Travel Information Service, *IPO Annual Progress Report*, Eindhoven, The Netherlands.