EXTENDING THE SUSI SYSTEM WITH NEGATIVE KNOWLEDGE

B. Vromans[1], *R.J. van Vark*[2], *B. Rueber*[3], *A. Kellner*[3]

[1] KPN Research, St. Paulusstraat 4, 2264 XZ Leidschendam, The Netherlands.
E-mail: b.vromans@research.kpn.com
[2] Delft University of Technology, Faculty of Inf. Techn. and Systems, Zuidplantsoen 4, 2628 BZ Delft, The Netherlands.
E-mail: r.j.vanvark@cs.tudelft.nl
[3] Philips GmbH Research Laboratories Aachen, Weisshausstr. 2, D-52066 Aachen, Germany.
E-mail: {rueber,kellner}@pfa.research.philips.com

ABSTRACT

The Philips SUSI system is a set of modules for the construction of automatic inquiry systems such as the train timetable information system TABA and the Philips Automatic Directory Information System (PADIS). These systems conduct a dialogue in continuous spontaneous speech with the user, giving access to information in a database. To improve the system behaviour we have extended the knowledge representation and update functions of the SUSI system so that they collect and use the negative knowledge during the dialogue. Negative knowledge in this context is the information that results from denials by the user in the dialogue, such as "No" and "Not Mr. Smiths". The negative knowledge handling has been implemented and tested for the SUSI based application PADIS. For PADIS, we found that certain dialogues improve significantly because of the use of negative knowledge.

INTRODUCTION

The Philips SUSI system is a set of modules from which the Philips Automatic Directory Information System (PADIS) [2] and TABA [1] automatic inquiry system have been constructed. In this paper we discuss the extension of the SUSI system with negative knowledge. The PADIS system used for call completion within Philips Research Labs Aachen, will be used to illustrate the effects of the changes in the SUSI modules. PADIS can handle spontaneous continuous speech and is a mixed-initiative dialogue system. In addition to call completion, PADIS can also give information about e-mail addresses, room numbers or telephone numbers. An example PADIS dialogue[1] is dialogue 1.

In this paper we focus on the negative knowledge that can be gathered from a PADIS dialogue and used by the system. Negative knowledge in this context is the information that results from denials by the user in the dialogue. Turn *U2* in dialogue 1 shows us such a denial.

[1] Most PADIS dialogues consist of one or two turns. The average concept recognition rate is 98%. In this paper we give examples of the lengthier dialogues with more recognition errors than usual to illustrate the effects of using negative knowledge.

S1: "PADIS – How can I help you?"
U1: "I would like to speak to Matthew."
S2: "Would you like to speak to Mr. Martens?"
U2: "No, Matthew!"
S3: "What's Matthew's last name?"
U3: "I don't know, but he works for the MI-group."
S4: "Matthew Harris. I'll connect you..."

dialogue 1: example PADIS dialogue

Previous implementations of PADIS only stored the *positive* knowledge gathered in the dialogue. The denials by the user were only used to remove elements from the positive knowledge. This could lead to the following situation (where the name "Rueber" is (mis)recognised by the system as "Gröber" in turns *U1* and *U3*):

S1: "PADIS – How can I help you?"
U1: "I would like to speak to Rueber"
S2: "Do you want to speak to Mr. Gröber?"
U2: "No, not Gröber."
S3: "Who would you like to speak to?"
U3: "Rueber"
S4: "Do you want to speak to Mr. Gröber?"
...

dialogue 2: recurrence of a recognition error

As can be seen in *S3* in dialogue 2, the system has processed the denial of Gröber by removing Gröber from the positive knowledge. In *S4* however, we can see that the system has 'forgotten' the fact that the user does not want to speak to Gröber and the system is unable to handle the recurring recognition error. As dialogues like dialogue 2 are clearly undesirable (see e.g. [4]), we investigated how to make the system use the negative knowledge in the dialogue.

We describe a method for applying negative knowledge that prevents the system's user utterance recognition process from recognising something that has already been denied by the user. In this method two separate activities are important (for a complete description see [6]):

- extracting negative knowledge from the user utterance and system prompt, and
- applying negative knowledge to improve user utterance recognition.

EXTRACTING NEGATIVE KNOWLEDGE FROM THE USER UTTERANCE AND SYSTEM PROMPT

The first step we make towards introducing negative knowledge in the PADIS system is to extract negative knowledge from the denials in the user utterance. As the user generally denies (information from) the system prompt, we have to take into account the system prompt for the extraction of negative knowledge. The negative knowledge extraction is demonstrated by using a semantical representation of the system prompt, *R*. The representation consists of a set of slot-value pairs, which are used in SUSI to represent semantical information.

S: "Do you want to speak to Smiths?"
R: {request: connect, name: Smiths}

figure 1: semantical representation of system prompt

We will now discuss the types of denials that are possible in PADIS, each with their own consequences for negative knowledge. As all incoming information in a system with a speech interface is more or less unreliable due to imperfect speech recognition and understanding, we also have to take into account the reliability of the concluded negative knowledge. We will discuss the just-"no" denial, the specified denial and the implicit denial.

Just-"no" denial

A just-"no" denial is found when the user answers "no" to a system prompt without specifying other denials. In this case we can only conclude that the user does not agree with what has been said or asked in the system prompt. However, the reliability of this information is high, as "no" is such a simple utterance that other user utterances are hardly ever mistaken for it by the speech recogniser. The following example illustrates the negative knowledge extraction for the just-"no" denial.

S2: "Do you want to speak to Mr. Gröber?"
R2: {request: connect, gender: male, name: Gröber}
U2: "No"
K-2: {request: connect, gender: male, name: Gröber}

figure 2: negative knowledge for just-"no" denial

Here *K-2* represents the negative knowledge extracted from dialogue turn 2. It is the set of slot-value pairs denied by the user. The system cannot conclude anything about the individual slot-value pairs, but it knows that the combination of these slot-value pairs is wrong, as it has been reliably denied by the user.

Specified denial

A specified denial is found when the user denies specific slot-values from the system prompt as in figure 3. In this case the negative knowledge that can be concluded for the dialogue turn consists of the set of slot-value pairs denied by the user. We assume that this is always a subset of the slot-value pairs in the semantical representation of the system prompt. This implies that the user only denies information appearing in the system prompt. For PADIS and other dialogue systems that have simple dialogues, this is a very reasonable assumption. For example it is not likely that the user responds to "Do you want to speak to Smith?" with "No, I don't want to speak to Jones." This assumption can be exploited to improve user utterance understanding, as is done in [5].

S2: "Do you want to speak to Mr. Gröber?"
R2: {request: connect, name: Gröber}
U2: "Not Mr. Gröber"
K-2: {gender: male, name: Gröber}

figure 3: negative knowledge for specified denial

Phrases like "not ..." and "don't want to ..." can also be recognised very reliably by the speech recogniser and as we only allow values from the system prompt to be denied, there is no possibility for the system to confuse the values. For example, if in figure 3 the system would have understood *U2:* "Not Mr. Rueber" instead of *U2:* "Not Mr. Gröber", this would have been ignored[2]. These considerations lead to the resulting negative knowledge being reliable, i.e. largely insensitive to speech recognition errors.

Implicit denial

Next to saying "no" or explicitly denying certain information, the user can also introduce negative knowledge without explicitly denying anything, as in figure 4.

S2: "Do you want to speak to Gröber?"
R2: {request: connect, name: Gröber}
U2: "Rueber"
K-2: {name: Gröber}

figure 4: negative knowledge for implicit denial

From figure 4 one might conclude that "Gröber" should be added to the negative knowledge. We however will not use the implicit denial for the extraction of negative knowledge, as the resulting negative knowledge is not reliable, being sensitive to small speech recognition errors. For example if the user in *U2* said "Gröber" and this was confused for "Rueber" then this would result in "Gröber" being added to the negative knowledge.

To summarise we extract negative knowledge from the user utterance and system prompt in case of a just-"no" denial and a specified denial. In these cases it is possible to obtain negative knowledge that is more reliable in general than new or old unverified positive knowledge as it is largely insensitive to speech recognition errors. This fact will be used in "applying negative knowledge to improve the interpretation of the user utterance".

For the extraction of the negative knowledge in the foregoing paragraphs we have used a semantical representation of the system prompt. However, in

[2] It is in fact removed from the *n*-best list, see under "applying negative knowledge to improve user utterance recognition"

PADIS we only have a textual representation and not a semantical one. Therefore, instead of using a semantical representation of the system prompt we will use the positive knowledge that has been gathered during the dialogue and use the same rules for the extraction.

Like the semantical representation of the system prompt the positive knowledge in the SUSI system is represented by a set of slot-value pairs. As an example dialogue 3 illustrates the build-up of the positive knowledge (denoted by *K*+) during dialogue 2.

K+: {request: connect}[3]
S1: "PADIS – How can I help you?"
U1: "I would like to speak to Rueber"
K+: {request: connect, name: Gröber}
S2: "Do you want to speak to Mr. Gröber?"
U2: "No, not Gröber."
K+: {request: connect}
S3: "Who would you like to speak to?"
U3: "Rueber"
K+: {request: connect, name: Gröber}
S4: "Do you want to speak to Mr. Gröber?"

dialogue 3: positive knowledge represented as slot-value pairs

The system prompt generated by the system is based directly on the positive knowledge. There may be things in the positive knowledge that are not in the system prompt. For example, in figure 5 by saying "No" the user does not refer to {gender: male} as it is not in the system prompt but it is included in the negative knowledge as the extraction is done using the positive knowledge. There can be however, no information in the system prompt that is not in the positive knowledge or contradicts with the positive knowledge. So the negative knowledge may be less *precise* than it would be if we would have a semantical representation of the system prompt, but it is not *wrong*.

K+: {request: connect, gender: male, name: Gröber}
S2: "Do you want to speak to Gröber?"
R: {request: connect, name: Gröber}
U2: "No."
K-2: {request: connect, gender: male, name: Gröber}

figure 5: using the positive knowledge instead of the semantical representation of the system prompt for negative knowledge extraction

As we can in general extract negative knowledge (like *K*-2 in figure 5) in every dialogue turn *K*-, the negative knowledge gathered in the dialogue, combines the negative knowledge from every dialogue turn.

[3] The system assumes connection (call completion) to be the default service. If the user mentions just a name when calling PADIS, the user is automatically connected to this person.

APPLYING NEGATIVE KNOWLEDGE TO IMPROVE USER UTTERANCE RECOGNITION

Now that we have described how to extract the negative knowledge from the user utterance and how to represent it, we will describe how the negative knowledge is applied to improve the SUSI user utterance recognition.

In the SUSI system each user utterance leads to an *n*-best list of concept paths, each representing a possible interpretation of the user utterance. A concept path is a list of slot-value pairs representing the semantical information in the user utterance. The *n*-best list of concept paths for *U1* in dialogue 3 could be:

1. {request: connect, name: Gröber}
2. {request: connect, name: Rueber}
...

In the *n*-best list the concept paths are sorted according to their acoustic and language model probability given the user utterance. In every dialogue turn certain concept paths, such as concept paths containing denials of values not in the positive knowledge, are discarded from the *n*-best list. The highest-ranking concept path is selected from the remaining concept paths in the *n*-best list. This concept path is then used to update the positive knowledge and therefore influences the next system prompt.

K+: {request: connect}
S1: "PADIS – How can I help you?"
U1: "I would like to speak to Rueber"
K+: {request: connect, name: Gröber}
"Rueber" has been recognised as "Gröber".
S2: "Do you want to speak to Gröber?"
U2: "No, not Gröber"
K+: {request: connect}
K-: {{name: Gröber}}
{name: Gröber} added to neg. knowledge.
S3: "Who do you want to speak to?"
U3: "Rueber"
C3,1: {name: Gröber}
This concept path is rejected as it would lead to pos. knowledge containing {name: Gröber}
C3,2: {name: Rueber}
Second best concept path. This is accepted.
K+: {request: connect, name: Rueber}
K-: {{name: Gröber}}
S4: "Do you want to speak to Rueber?"
U4: "Yes"
S5: "Connecting to Rueber..."

dialogue 4: cancelling the effect of the recurring recognition error from dialogue 2.

By using the negative knowledge for the process of discarding concept paths, we can discard concept paths that lead to positive knowledge containing information that has already been denied by the user. This is achieved by discarding from the *n*-best list every concept path that would result in positive knowledge containing a

superset of one of the sets in the negative knowledge. (Similarly in [3] negative knowledge is used to make 'negative predictions' to improve recognition of the user utterance).

The examples in dialogue 4 and dialogue 5 show this. In these examples for dialogue turn *i*: *K+* is the positive knowledge, *K-* is the negative knowledge and *Ci,j* is the j^{th} concept path in the *n*-best list of concept paths in turn *i*. The first example (dialogue 4) shows how the application of negative knowledge cancels the effect of the recurring recognition error from dialogue 2.

An example of negative knowledge consisting of more than one set in a longer dialogue is shown in dialogue 5. Some of the system prompts differ from the ones used in the current PADIS version. This is done to achieve a better illustration of the use of negative knowledge in the dialogue.

K+: {request: connect}
S1: "PADIS – How can I help you?"
U1: "The e-mail of Smith."
Recognised as "Samuel Smith."
K+: {request: connect, fname: Samuel, name: Smith}
S2: "Do you want to speak to Samuel Smith?"
U2: "No, not Samuel Smith"
K+: {request: connect}
K-: {{fname: Samuel, name: Smith}}
S3: "Who do you want to speak to?"
U3: "Not speak, e-mail!"
Recognised as "Not speak, Samuel."
K+: {fname: Samuel}
K-: {{request: connect}, {fname: Samuel, name: Smith}}
The neg. knowledge now consists of two sets.
S4: "What do you want of Samuel?"
In a real dialogue, this is not a feasible question.
U4: "No!"
K+: {}
In a system without neg. knowledge, we would have been back where we started, but the gathered neg. knowledge will help the system to prevent making the same errors once more.
K-: {{request: connect}, {fname: Samuel}}
The negative knowledge set {fname: Samuel, name: Smith} has been removed because {fname: Samuel} has been added.
S5: "PADIS – How can I help you?"
U5: "The e-mail of Smith."
C5,1: {fname: Samuel, name: Smith}
This concept path is rejected.
C5,2: {request: e-mail, name: Smith}
K+: {request: e-mail, name: Smith}
K-: {{request: connect}, {fname: Samuel}}
S6: "The e-mail of Smith?"
U6: "Yes!"

dialogue 5 : negative knowledge in a longer dialogue

CONCLUSION

The SUSI modules have been extended with functions for negative knowledge representation and handling. Negative knowledge is represented as a set of sets of possibly denied slot-value pairs. The method of extracting the negative knowledge guarantees that only relatively reliable negative knowledge is gathered. The negative knowledge is used to discard concepts paths that would lead to positive beliefs of which parts have already been denied by the user. This can prevent the dialogue from getting stuck because of the same recognition error occurring over and over again.

Though no quantitative results are available yet, tests have shown that dialogues where the system confuses two or more similar sounding names (such as dialogue 2 and dialogue 4) have improved greatly. In other PADIS dialogues no large improvements are to be expected, as typically dialogues are too smooth and short to contain denials.

REFERENCES

[1] Aust, H., Oerder M., Seide F., and Steinbiss V. (1995), The Philips automatic train timetable information system. *Speech Communication*, 17(3-4):249-262.

[2] Kellner, A., Rueber B., Seide F., and Tran B. (1997), PADIS – an automatic telephone switchboard and directory information system. *Speech Communication*, 23(1-2):95-111.

[3] McGlasham, S. (1996), Towards multimodal dialogue management. In: Luperfoy, S., et al. (eds.), Proc. TWLT 11: Dialogue Management in Natural Language Systems, University of Twente, pp. 13-22.

[4] Rothkrantz, L.J.M., Manintveld W.A.Th., Rats M.M.M., Vark R.J. van, Vreught J.P.M. de, and Koppelaar H. (1997), An Appreciation Study of an ASR Inquiry System, EuroSpeech '97, Rhodes, Greece, pp. 1715-1718.

[5] Seide, F., Rueber, B., Kellner, A. (1996), Improving speech understanding by incorporating database constraints and dialogue history. In: Proc. ICSLP, volume 2, pp. 1017-1020, Philadelphia, PA.

[6] Vromans, B. (1998), Using history in dialogue management of automatic inquiry systems. Master's thesis, Delft University of Technology, The Netherlands.