Recognition Error Handling in Spoken Dialogue Systems

Genevieve Gorrell

Linköping University*

Abstract

Increasing use of mobile devices for information access brings with it a demand for robust speech recognition. An important way in which speech recognition system performance can be improved is through use of dialogue strategies to handle the situation in which the system fails to recognise the user's utterance. In the work described here, the case is made for combining multiple speech recognisers and appropriate dialogue strategies to handle poor recognition results. Two implemented systems are discussed as examples.

1 Introduction

Use of mobile and ubiquitous multimedia in the form of small informational devices, along with an increased interest in informational access during parallel activities brings speech recognition very much into focus. Firstly, mobile devices present challenges for data input, and speech is a very natural option for human beings. Secondly, combination of information access with other activities often requires the freeing of the hands, precluding many data entry options. At the same time, environments in which mobile devices might be used are challenging for speech recognition; noisy environments and distracted users both place demands on performance.

Consider an interactive informational dialogue system, such as a telephone system making train times or public information available, or selling cinema or airplane tickets. Speech recognition presents challenges for dialogue in that the possibility must always be accounted for that the system misrecognised what the user said. A spoken dialogue system is in this way quite different to a written one. Speech enabling an existing text-based system therefore involves more than just adding a speech-recogniser onto the front; a successful integration will involve some changes to the dialogue structure.

To introduce spoken dialogue into the mobile environment, a lot of work needs to be done on making such systems robust to error. Much work has been and is being done on error handling in spoken dialogue, both in the research sector and commercially, see for example [Err 2003]. Errors can be recovered from after the fact; a recognition result already accepted by the system can be corrected by the user after some implicit confirmation/feedback to the user about what the system believed was said. For example;

System: Destination?
User: Linköping
System: And how would you like to travel to Nyköping?
User: No, I said Linköping

More explicit confirmation can take the form of an entire dialogue turn devoted to telling the user what was understood and asking them if the understanding was correct; "Nyköping, is that correct?" The most appropriate strategy depends on the context - how critical is it that the system understood correctly - and also, how much doubt is there about the interpretation.

Examples of situations where a system mistake can be high-cost include home control systems, where we wouldn't want for example the freezer to be turned off by mistake, or ticket booking, where again we wouldn't want a bank account to be debited incorrectly. Low-cost situations might include question-answering systems, where it is as quick to answer the wrong question as it is to confirm the understanding of the question.

There are various ways in which the system can glean information about the reliability of a recognition result. For example, a low confidence score from the speech recogniser would be an obvious indicator that some confirmation or disambiguation would be advisable. Another source of information is the likeliness of that utterance given that context. An utterance might be irrelevant to the domain, semantically meaningless, or just simply beyond the scope of the system.

This work discusses the situation where either the user says something beyond the scope of the system, or the utterance is recognised incorrectly as something beyond the scope of the system. Two implemented approaches will be described. The first, Targeted Help [Gorrell et al. 2002] focuses on training the novice user. The second develops on the first, and is equally appropriate to the more experienced user, who perhaps is having difficulty getting recognised by the system due to the acoustic environment.

The next two sections will describe each approach in detail, before the two approaches are discussed in comparison to each other in the following section. The conclusion will focus on the context of the work and future directions.

2 Description of the First Approach

User: Hello?
System: Hello. How can I help?
User: Um, can I have some lights on?
System: Sorry, I didn't quite catch that. To turn a device on or off, you could try something like "turn on the kitchen light".
User: Turn on the kitchen light.
System: The kitchen light is now on.
User: What other sorts of things are there?

*email:gengo@ida.liu.se

System: I didn't quite catch that. To find out what devices there are, you could try something like "what devices are there", or "what devices are there in the lounge".
User: What devices are there in the lounge?
System: There are three things in the lounge; the computer, the vcr and the television.
User: Turn on the tv and the vcr and check the computer is off.
System: I didn't quite catch that. Long commands can be difficult to understand. Perhaps try giving separate commands for each device.
User: Okay, check the computer's off ...

When a dialogue system fails to obtain a recognition result with sufficient confidence, the usual way to handle it is to discard the result, perhaps ask the user to repeat themselves or change dialogue strategy. This is despite the fact that a poor recognition result can contain useful information.

It is also usual that some general help is available in how the user should address the system. This help is typically available whenever the user asks for it, for example, by saying "help", and is also perhaps returned automatically to the user after three recognition failures in a row. The help, being general to the system, can be quite unwieldy and tedious to listen to.

The Targeted Help approach features a number of more specific help messages, each helping the user to perform a particular task. When the user fails to get recognised, their utterance is used to choose the help message most likely to assist them in doing what they appear to be trying to do. This message is returned after each recognition; being specific and fairly brief, the main disincentive to giving help immediately on recognition failure is reduced. Subject trials show improved performance with the Targeted Help system compared with a standard strategy.

2.1 Description of the system

The base system is the On/Off House (OOH) system, a smart home system implemented using the Nuance Toolkit platform [Nuance 2002]. It offers English spoken language control, via telephone, of about 20 devices in a simulated home. Device types include both on/off and scalar. The dialogue manager is implemented in Visual C++ using the Nuance DialogueBuilder API. The mode of operation is primarily user-initiative, that is to say, the dialogue is led by the user. The grammar offers coverage of a fairly broad range of language, including commands ("Turn on the heater", "Turn off the light in the bathroom") and several types of questions ("Is the heater switched on?"; "What is there in the kitchen?"; "Where is the washing machine?"; "Could you tell me which lights are on?").

Targeted help has been added to the system such that whenever an utterance is not recognised above a certain confidence threshold using the grammar-based system some further processing is done. First, the utterance is passed to a domain-specific recogniser based on a statistical language model (SLM) for a second recognition. The result of this recognition contains information such as what words the recogniser recognised, what confidence scores it places on those words, what confidence score it places on the entire utterance, etc. This result is used to create a feature set. A decision tree classifier is then used to classify the feature set and return the class. This class maps to an error message, which is played to the user before returning to the main loop of the application. The error message played will typically be of the generic form

> "I didn't quite catch that. To *(carry out some action)*, you could try something like *(example of suitable command)*."

Section 2.3 contains examples of error messages.

The SLM, described in more detail in [Knight et al. 2001], was created using about 4000 transcriptions of utterances collected using the On Off House system, plus a further 200 utterances appropriate to the home control domain collected using only recognition feedback. The performance of this SLM is comparable to that of the grammar-based recogniser over a mixed corpus.

2.2 Classifier

The module responsible for selecting the help message has been implemented as a simple decision tree classifier built using the popular See5 system [RuleQuest 2002]. This classifier was trained on about 1000 utterances, including the 200 less constrained utterances used to create the SLM. The remainder of the classifier training corpus was also taken from the SLM training corpus. The training data was prepared by recognising each utterance in turn using the SLM and then processing the output of the recognition to produce the desired feature set. This means that both at training time and at run-time the classifier is using output of the same SLM.

The classifier feature set used consisted of the following: the individual words; their confidence scores; the utterance confidence score; the number of words in the utterance; the number of occurrences of each of the items "on/off", "the", "and" and "turn/switch" (four features); whether or not the utterance started with each of the items "what is there/what's in", "is there", "where/where's", "turn/switch", "the", "what is on/what is switched on/what's on", "which", "could/can/would", and "are/is" (nine features); whether or not the utterance contained an occurrence of each of the items "are there any/is there any", "what/what's", "is anything", "turn/switch", "please" and "everything/all" (six features); and whether the utterance ends with "are there".

2.3 Classes

The classifications used were hand-selected based on observation of the corpus. Below are the most common of the 12 classes with their associated error messages and percentage of the training corpus covered by each;

REFEXP_COMMAND(35%) - "I didn't quite catch that. To turn a device on or off, you could try something like 'turn on the kitchen light'."

LONG_COMMAND(13%) - "I didn't quite catch that. Long commands can be difficult to understand. Perhaps try giving separate commands for each device."

PRON_COMMAND(11%) - "I didn't quite catch that. To change the status of a device or group of devices you've just referred to, you could try for example 'turn it on' or 'turn them off'."

REFEXP_STATUS_QUERY(9%) - "I didn't quite catch that. To find out the status of a device, you could try something like 'is the light on' or 'is the kitchen light on'."

DEFAULT_ERROR(15%) - "Sorry, try again."

2.4 Decision Tree

The baseline error rate for the classification task is 65%, if the system classifies everything as a REFEXP_COMMAND (the most common class). Classification based only on the first word improves the error rate to 40%. The error rate for the final decision tree, measured using cross-validation on the training data, was 12.2%.

2.5 Evaluation

User trials were performed in order to determine the success of the approach. The targeted help system and a control system were made available over the telephone. Both systems made the user aware of the global "help" option in the introductory prompt. The global help message was the same in both systems and consisted of all the targeted help messages read out in sequence. Upon recognition failure, the "helpful" system issued a targeted help message as described above, whereas the control simply had one behaviour: first, to say "Sorry, try again" and then, on a consecutive failure, to issue a short version of the initial help message. This control strategy was chosen to reflect an approach commonly taken in commercial systems.

31 novice users were given a scenario which involved ringing a voice controlled house and leaving it in a secure state: fire risks were to be minimized, yet the house should appear occupied to deter burglars. After completing the task, users filled in a short questionnaire. Sixteen users performed the task with the targeted help system, and fifteen with the control. The following were measured; task completion, dialogue length, word error rates, in coverage rates, and users' knowledge of the house and the system's capabilities.

Analysis showed that users of the targeted help system had a significantly greater average task completion score, and their word error rates were very significantly lower than the control group's. The number of in-coverage utterances was also significantly higher in the targeted help system. This suggests users more quickly attune themselves given the intelligent help system. The word error rates for our statistical language model show a similar pattern over our corpus, being notably lower for the targeted help users. For a more detailed description of the results, see [Gorrell et al. 2002].

Users of the intelligent help system also used a greater variety of constructions in their interactions. For example, as well as "turn on the X" and "is the X on" they were more likely to try "what is on" or "what devices are there". Also, after hearing a system suggestion, they often used it immediately. Questionnaire results suggested that Targeted Help users were ending the calls with a greater awareness of the state in which they left the house. Examination of the conversations also suggest a more systematic approach to checking the house on the part of the Targeted Help users.

2.6 Discussion of the First Approach

The positive evaluation result suggests that there is scope for improving system performance with regards to recognition failure behaviour in spoken dialogue systems using the techniques outlined here. Based on this, the system described in the next section develops the theme.

3 Description of the Second Approach

User: Where does the swan go to in the winter? (recognised correctly)
System: Do you mean "Where does the swan migrate to"?
User: Yes
System: The swan migrates to ...
User: What about the seagull? (recognised as "water seagull")
System: Do you mean "how about the seagull"?
User: Yes
System: The seagull migrates to ...

In Targeted Help, twelve messages were designed to broadly cover all the things that the user might want to do with the system, and a classifier was used to attempt to deduce, from a statistical recognition result, which they were aiming at. The approach proved robust and was well-received by the test group. However, use of the decision tree classifier limited the number of help messages that could be offered; the more messages, the more mistakes the classifier will make. In fact, the interpretation grammar provides the ultimate definition of the range of utterances that the system can handle, from a natural language understanding perspective, and is much finer-grain than the approximately 12 categories that Targeted Help was limited to.

In this approach, a first step is taken toward using the interpretation grammar to assist the user in getting the desired response from the system. The dialogue is also streamlined, compared with Targeted Help; in the case where the system guesses correctly, the user is not required to repeat the suggestion.

The process, summarised, is as follows. Two grammars are used to recognise the user's utterance; a context-free grammar and a trigram language model. The trigram is used as a back-off, so a result accepted with a confidence greater than 45 on the CFG is not passed on to the trigram. The result is then parsed with a guide grammar. If the utterance does not parse, then a suggestion is formulated using the guide grammar and returned to the user. (If the result does parse, it is accepted by the system.)

3.1 Description of the System

A speech recognition component has been added to the BirdQuest system [**?**]. BirdQuest is a question-answering system that answers questions in Swedish about Nordic birds.

The recogniser makes use of Nuance 8, and consists of a hand-coded CFG grammar, plus a back-off statistical language model. The grammar-based recogniser has been in use since January 2003 and has been through some three development iterations. It achieves a word error rate of 56% on a transcribed corpus of 633 utterances collected using it. This is admittedly rather high but the corpus is small and contains a disproportionate amount of non-native Swedish speech. Also the domain is such that questions of very varied form appear. It includes many novice users and much of the data was collected in noisy conditions. The SLM was created from this corpus. The small size of the SLM training corpus leaves the reliability of the SLM open to question, and also precludes the separation of a test corpus for demonstration of its performance. However, anecdotally, performance is sufficient to demonstrate the technique that is the main point of this work. Combined, the recognisers perform surprisingly well, and it is hoped that both recognisers can be

improved upon in the future.

Speech recognition is implemented in such a way that a result from the grammar-based recogniser with a confidence score over 45 is accepted without question. An utterance rejected by the grammar-based recogniser is passed on to the SLM, and recognised with a confidence threshold of 30. If the utterance is not accepted by either recogniser, then the system simply rejects the utterance; "I'm sorry, try again". If however the utterance is accepted, then the following further processing is done.

A guide grammar has been created, closely based on the CFG used in the primary recogniser. Every result is first parsed using this grammar. If the utterance parses then the system accepts it. If, however, the result does not parse using the guide grammar, then the closest string in the grammar to the utterance is returned to the user as a suggestion; "Did you mean X?" The user can reply yes, in which case the suggestion is fed into the system. Or they can reply no, in which case the system apologises. Or they can ignore the suggestion entirely and simply say a new question (or repeat the old one).

3.2 The Grammar

The grammar is a hand-coded CFG in Nuance GSL format. It is ad hoc, not particularly linguistically-motivated, though an attempt has been made to enforce agreement constraints, and includes some hand-coded probabilities, which have been demonstrated to make a performance improvement of a couple of percent. It contains some 55 rules in total.

3.3 The Corpus

The corpus comprises three collections of utterances in which external users were allowed access to the system, plus a collection of material acquired during development and internal trials. It consists of a transcribed set of 633 wave files. The first of the collections performed with external users was obtained during a conference demonstration. At this stage, recognition was done using a smaller "first cut" grammar. There is noticeable background noise in this section of the corpus. Users were given minimal guidance. The second external data collection was performed during a university open day, and is similar in nature to the first. The third collection was acquired during a demonstration to a small group. At this stage, the grammar had been improved and the statistical recogniser had also been added to the system. The background noise is significantly reduced in this data collection. Additionally, the users were slightly familiar with the system, having had the opportunity to try it during the first external data collection. The majority of the utterances collected during development were spoken by the author, and are somewhat limited in coverage. Other speakers do also feature however.

Additionally, a quantity of written domain-appropriate material was available from a corpus collection performed earlier to inform the development of the original text-based system. This corpus has also been utilised in creating the recognition language models.

3.4 The SLM

Transcriptions of the entire spoken corpus, plus the written corpus, were compiled, using Nuance 8 tools, into a 3-gram statistical language model. Performance figures are not available at this stage because the corpus is too small to allow for a test section to be separated out. Anecdotally, however, the performance is comparable to or perhaps marginally better than that of the grammar, though the nature of the utterances on which good results are obtained differs.

3.5 Forming Suggestions - The Parser

The recognition result is processed in the following way. First an attempt is made to parse the result with the "guide grammar", a grammar similar to the recognition grammar but designed for this purpose. If this parse succeeds then the result is taken to be a good one. If the parse does not succeed then every path through the guide grammar is assessed for its closeness to the result. Closeness is gauged in terms of number of shared words. The best path, ie. the sentence allowed by the grammar that has most words in common with the recognition result, is returned to the user as a suggestion.

3.6 Discussion of the Second Approach

The second approach is intended as a development of the first in a number of ways; a finer grain of assistance is given, and the dialogue is streamlined and made appropriate for more experienced users. Whilst the work is currently unevaluated, initial impressions suggest it is well-received. The next section discusses both approaches in the context of comparable work and the thesis of this discourse.

4 Discussion of Both Approaches

We have described two systems that combine grammar-based and robust approaches to natural language understanding by using robust methods to assist the user in the case where their recognition result is poor. The first is applied to a command-and-control system, and the second, to a question-answering system. Evaluation of the first system has shown positive results in terms of user's increased ability to get recognised by the system and to accomplish a task. It is worth remembering that our sample was restricted entirely to people who had never used the system before. The second system develops on the first, allowing a finer grain of assistance to be given, as well as streamlining the dialogue and making in appropriate for more experienced users.

This work is intended to make the case for combining multiple speech recognisers and appropriate dialogue strategies to handle poor recognition results. Both the suggestions outlined here make use of recognition results that would normally be discarded; the first by using a second recogniser where the primary recogniser fails, and then using that result to select a help message for the user, and the second, by using a recognition result that the system is unable to handle to again provide some assistance, this time in the form of a suggestion. There is no reason why these two approaches, along with other similar ones, could not be combined in the same system. There are enormous possibilities for improving dialogue behaviour in the case of a poor recognition result.

Both the approaches outlined here use multiple recognisers; specifically, the differences in strengths between grammar-based and statistical language modelling are exploited. In the first, the statistical recogniser is specifically used to inform the selection of the help message returned to the user. In the second, the statistical recogniser is used as a back-off to the grammar-based recogniser, and therefore

forms part of the main recognition strategy rather that just a part of the recognition failure handling strategy, though in practice, it is a result rejected by the grammar and accepted by the statistical recogniser that is most often responded to with a suggestion; a result accepted by the recognition grammar often parses with the guide grammar and so is accepted without further processing. (One may wonder why the guide grammar and the recognition grammar differ at all; recall that the guide grammar reflects the system's abilities, whereas the aim in creating a speech recogniser is to recognise as much of what the user says as possible. Therefore the system may on the one hand not know what the user said, or on the other hand, know but be unable to assist. This distinction is evident in human-human communication and can very well be made in human-machine communication as well.) For a more in-depth discussion of the differing strengths of grammar-based and statistical language modelling for speech recognition, see [Knight et al. 2001]. For a further suggestion on how these differing strengths can be exploited, see [Gorrell 2003].

Future directions will include quantitative demonstration of the success of the second approach described here. Work remains to be done in developing the strategy used to select the suggestions. Furthermore, combination of these approaches and other similar ones in one multimedia system is an appealing next step.

References

2003. *Proceedings of Error-Handling in Spoken Dialogue Systems*. Eurospeech satellite event.

Gorrell, G., Lewin, I., and Rayner, M. 2002. Adding intelligent help to mixed initiative spoken dialogue systems. In *Proceedings of ICSLP*.

Gorrell, G. 2003. Using statistical language modelling to identify new vocabulary in a grammar-based speech recognition system. In *Proceedings of Eurospeech*.

Knight, S., Gorrell, G., Rayner, M., Milward, D., Koeling, R., and Lewin, I. 2001. Comparing grammar-based and robust approaches to speech understanding: a case study. In *Proceedings of Eurospeech 2001*, 1779–1782.

Nuance. 2002. http://www.nuance.com. as of 15 March 2002.

RuleQuest. 2002. http://www.rulequest.com. as of 15 Mar 2002.