ADDING INTELLIGENT HELP TO MIXED-INITIATIVE SPOKEN DIALOGUE SYSTEMS

Genevieve Gorrell, Ian Lewin and Manny Rayner

Fluency Voice Technology
Westbrook Centre
Milton Road
Cambridge CB1 1YG
United Kingdom

{genevieve.gorrell, ian.lewin, manny.rayner}@fluencyvoice.com

ABSTRACT

The rapidly expanding voice recognition industry has so far shown a preference for grammar-based language modelling, despite the better overall performance of statistical language modelling. Given that the advantages of the grammar-based approach make it unlikely to be replaced as the primary solution in the near future, it is natural to wonder whether some combination of the two approaches may prove useful. Here, we describe an implemented system that uses statistical language modelling and a decision-tree classifier to provide the user with some feedback when grammar-based recognition fails. Users of this system had more successful interactions than did users of a control system.

1. INTRODUCTION

Over the last few years, continuous speaker-independent speech recognition technology has reached the point of commercial viability, and with this landmark has come an explosion of interest from industry in spoken dialogue systems [1, 2, 3, 4, 5]. Interestingly, the approach taken by the new commercial implementers differs significantly from that pursued over several decades in academia, particularly in the area of language modelling. The standard formula previously was to collect a corpus of utterances appropriate to the domain using Wizard of Oz simulations and use this to train a statistical language model. This formula produced some impressive systems [6, 7, 8, 9] and whilst there was speculation about the alternatives it was clear what the mainstream was.

Despite this, commercial implementers are now focusing almost exclusively on grammar-based language models, for a number of reasons. Collecting a large enough corpus to create an SLM is time-consuming and expensive, whereas a grammar can be written quickly. With SLM-based recognition, a separate parser must be implemented to extract semantic content, whereas a grammar-based language model can be annotated with semantics and double as a parser. Last but not least, grammar-based recognition can often perform better if users know what they can say to the system [10]. Commercial systems are typically system-initiative anyway, and if only a limited range of responses needs to be covered a grammar is the obvious choice.

[10] also demonstrated, however, the increased robustness of SLMs on unusual and less constrained utterances. Grammar based recognisers are brittle with respect to out-of-coverage utterances; in contrast, the performance of an SLM degrades much more gracefully. When user speech tends to be unconstrained, in particular in the case of novice users who simply don't know what they can say, an SLM may be a better choice. This separation of strengths suggests the idea of somehow combining the two approaches. If we try to do this, our basic goal will be to retain the advantages of grammar-based recognition for experienced users, while shading off into SLM-based recognition when users stray outside the coverage of the grammar or when grammar-based recognition fails for other reasons.

In the work reported here, a direct continuation of [10], we have attempted to implement a strategy of this general type. Our first idea was simply to recognise each utterance using both a grammar- and an SLM-based model, and then use the associated confidence scores to decide which result to pass on to downstream processing. Initial experiments were however unconvincing. For core domain utterances, measures based on confidence scores failed to discriminate accurately, and it was not obvious how to identify cases where the SLM-based result was more reliable than the grammar-based one. For less constrained utterances, neither recogniser did very well; even though the SLM-based one was usually much better, its output was typically still too noisy to be suitable for normal processing.

Examination of these results however suggested a more promising strategy. Instead of thinking of the SLM-based recogniser as a simple backup or alternative to grammar-based processing, we instead decided to use it to improve the quality of the help messages produced by the system when grammar-based recognition failed. Specifically, we implemented a small set of help messages, each of which was designed to be appropriate to a certain type of user utterance. When the confidence score returned by grammar-based recognition falls under a given threshold, the dialogue manager backs off to the SLM-based recognition result, and attempts to use its content to choose a relevant help message. Since the small space of help messages is far more constrained than the large space of structured semantic representations produced by the normal grammar, the task is correspondingly simpler, and can be attempted more realistically using the noisy input available.

We will call this approach *targeted help*. We describe here a series of experiments in which we have evaluated the utility of

We would like to thank Beth Ann Hockey for several helpful discussions.

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?

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 ...

Fig. 1. Example session with targeted help version of the home control system from [10]

the targeted help method in the context of the medium-vocabulary command and control task from [10]. We stress that in targeted help the SLM is not being used to improve the performance of the *recogniser*. On the contrary, we are trying to improve the performance of the *user*; the aim is to get naive users saying things entirely within grammar coverage as quickly as possibly, since this is ultimately the best way to optimise performance. The help messages are designed to provide feedback to users who are not recognised in such a way that the likelihood of their being recognised correctly in subsequent utterances is increased. Put slightly differently, the targeted help module's job is to guess roughly what the user is trying to do, and educate them with an in-coverage utterance that would achieve a similar end. Figure 1 shows a sample session with the targeted help version of the system.

The rest of the paper is organised as follows. Section 2 describes the base system, and Section 3 the targeted help module. Section 4 describes a simple evaluation, which contrasts the behaviour of naive subjects on targeted help and plain versions of the system. Section 5 concludes.

2. BASE SYSTEM

The base system is the On/Off House (OOH) system, which is implemented using the Nuance Toolkit platform [1], and 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. The grammar offers coverage of a fairly broad range of language, including commands ("Turn on the heater", "Turn off the light in the bathroom"), 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?"), universal quantification ("Switch off everything in the bathroom"), conjunction ("Are the hall and kitchen lights switched on?"; "Switch off the radio, TV and computer"), ellipsis ("Turn on the cooker"... "now the microwave") and pronouns ("Switch off the stereo and the hi-fi"... "switch them on again"). The system has been tuned over four or five iterations of user testing, and performs well enough to have been successfully demonstrated in public on several occasions.

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, instead of the standard error message, "Sorry, try again," being played to the user, some further processing is done. First, the utterance is passed to a domain-specific SLM-based recogniser 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 3.3 contains examples of error messages.

The SLM, which is described in more detail in [10], 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.

3. THE TARGETED HELP MODULE

3.1. Classifier

The module responsible for selecting the help message has been implemented as a simple decision tree classifier built using the popular See5 system [11]. 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. This means that despite the fact that at run-time targeted help only handles utterances rejected by the grammar, at training time a balanced sample of rejected and accepted utterances is used. This does mean that the training corpus was not exactly reflective of the run-time input the classifier would receive, but it enabled us to maximise the available training data, which we considered more important. The utterances were classified by hand using the transcriptions. 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.

3.2. Features

The first set of classifier features tried consisted only of the words and their confidence scores. Inspection of the decision tree produced by the classifier suggested that indirect approaches were being used to access certain information; for example, the presence of a fifth word might be made use of, probably as a way of determining whether the utterance contained less than five words. At the same time, features were being made use of that were unlikely to be anything more than a spurious feature of the data. This suggested that explicit inclusion of certain features, for example, number of words, might improve performance.

The final feature set used consists 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".

3.3. 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."

3.4. 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%.

4. EVALUATION

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 simply to reflect an approach commonly taken in commercial systems. Our objective was not to measure success against the best possible non-targeted help system (we know too that ours is not the best possible targeted help system) but rather against a plausibly representative one.

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. The only prior information given to users about the house was a small unlabelled floor plan with pictures of some furniture items (e.g. sofas, sinks, toilets) in order to indicate the number and type of rooms in the house. Users were not told which voice controllable devices were in the house (or whether they were on or off), what those devices could do, or any guidance as to how one should talk to the house. The idea was that knowledge about the voice control system should result entirely from experience with the system. After completing the task, users filled in a short questionnaire, designed to elicit, amongst other things how good a mental picture users had managed to build of the house, its devices, the operations they could perform and the linguistic capabilities of the speech interface. Sixteen users performed the task with the targeted help system, and fifteen with the control.

We measured dialogue length, word error rates, in coverage rates, and users' knowledge of the house and the system's capabilities.

Analysis showed that the word error rates of users of the targeted help system were very significantly lower than the control group's (39% versus 55%, for a one-tailed test $0.0002 > P$; with $\alpha = .999$ the difference in proportions lies between 10.5% and 21.3%). The number of in-coverage utterances was also significantly higher in the targeted help system (47% compared to 36%, for a one-tailed test $P = 0.0012$; with α=.90, the difference in proportions lies between 5% and 17%). Although both word error rates are high, this is to be expected in our scenario where users are completely new to the system and have received no guidance in how to talk to it. Furthermore, in the control system the error rate in the first five utterances is particularly high (76%) compared to the intelligent help system (45%). 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: 30% (for intelligent help) versus 44% (for control) overall; 40% versus 62% for the first five utterances. These rates are still high and the SLM does outperform the grammar based recognizer overall — but the important feature for our purposes [10] is the valuable output obtained on out of coverage utterances, for which the grammar based recognizer fails completely.

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; all bar three did this at some point and one user did it three times. In the control system, a user only has the welcome message or a subsequent "help" request to remind them of system capabilities. To our surprise, intelligent help system users actually requested help more often (only one control user requested help whereas six helpful users did). Our initial hypothesis was that control users would request more help because they would need it more. It may be that the intelligent help system actually helps users remember that useful help is available.

Some of our users, in both systems, showed a tendency to speak to the system in a robotic manner, for example, "turn on light" rather than "turn on the light." Such speech also displayed disfluency, hyperarticulation and voice raising, all of which negatively affect performance. As a crude measure of this, we simply counted the number of occurrences of the word "the" in the user's speech. Users of the intelligent system averaged 0.66 occurrences per utterance, whereas control users averaged 0.53. Furthermore, the conversations of robotic users reveal that whilst they were slow to extinguish this behaviour, a strategy of copying an intelligent help suggestion reduced the robotic speech tendency. (Users occasionally imitated even the suggestion's prosody).

Dialogues with the intelligent help system proved to be longer on average than those with the control. Only one dialogue, which was in the control, consisted of a total failure to achieve anything. Almost one-third (5) of the intelligent help assisted dialogues were longer than 35 turns, while none of the control dialogues were. Although the sample is small, three of those reveal users being guided to the correct way to express a question and then systematically exploring the house by asking questions ("What is there in the kitchen?") and then acting on what they were told about. In the control, only one dialogue of the longest five revealed even an attempt at systematic exploration. In both systems, there are good examples of systematic exploration when recognition accuracy is generally good. We had hoped our questionnaires would reveal users who felt in control of their dialogues (but see below).

Although the questionnaires indicated that intelligent help users ended their calls with a greater awareness of the state in which they left the house, the result was not statistically significant. We calculated awareness by scoring -1 for an incorrect statement about a device state and -0.5 for an "I don't know". (Intelligent help users averaged -2.6 and the control group averaged -3.6).

There was no significant difference in users' perception of the system's abilities. By scoring 1 for a correct answer (e.g. "yes" to "the system can turn on two devices at the same time") and -1 for an incorrect answer, the control group actually marginally outscored intelligent help users. However, it appears that perceptions bear only a passing semblance to reality. For example, more than half of the control users stated that they thought the system could understand a pronoun in "Switch it on" although there was only one instance of this actually happening! Possibly, users with very limited examples of successful interactions were just more likely to guess that other interactions they never considered were more likely to succeed.

5. CONCLUSIONS

We have described a system that combines grammar-based and robust approaches to natural language understanding by using a robust method to guide the user into coverage of a grammar-based recogniser. Initial evaluation of the 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. Giving the system a way to adjust its help strategy to the level of experience of the user is an interesting topic for future research.

There are obvious relationships between the work described here and the literature on call-routing [12, 13, 14]. In both cases, the central problem is to develop robust methods which can classify a spoken utterance into one of a set of possible alternatives and take appropriate action based on that classification (in this case returning an appropriate help message). Decision tree classifiers tend to overfit data when applied to unsuitable domains [15], and the general consensus is that latent semantic analysis and other matrix-based methods give best performance on call-routing tasks. We intend in the near future to implement a new version of our system using this kind of technology.

6. REFERENCES

[1] Nuance, http://www.nuance.com, 2002, as of 15 March 2002.

[2] SpeechWorks, http://www.speechworks.com, 2002, as of 15 March 2002.

[3] TellMe, http://www.tellme.com, 2001, as of 15 March 2002.

[4] BeVocal, http://www.bevocal.com, 2001, as of 15 March 2002.

[5] HeyAnita, http://www.heyanita.com, 2001, as of 15 March 2002.

[6] R. De Mori and R. Kuhn, "Some results on stochastic language modelling," in *Proceedings of the Speech and Natural Language Workshop*, Pacific Grove, CA, 1991, pp. 225–230.

[7] R. Rosenfeld and X. Huang, "Improvements in stochastic language modeling," in *Proceedings of the Speech and Natural Language Workshop*, Harriman, New York, 1992, pp. 107–111.

[8] W. Ward and S. Issar, "The CMU ATIS system," in *Proceedings of the ARPA Workshop on Spoken Language Technology*, Austin, Texas, 1995, pp. 249–251.

[9] M. Cohen, Z. Rivlin, and H. Bratt, "Speech recognition in the ATIS domain using multiple knowledge sources," in *Proceedings of the ARPA Workshop on Spoken Language Technology*, Austin, Texas, 1995, pp. 257–260.

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

[11] RuleQuest, http://www.rulequest.com, 2002, as of 15 Mar 2002.

[12] A. L. Gorin, G. Riccardi, and J. H. Wright, "How may I help you?," *Speech Communication*, vol. 23, no. 1/2, pp. 113–127, 1997.

[13] G. Riccardi, A. Gorin, A. Ljolje, and M. Riley, "A spoken language system for automated call routing," in *Proc. ICASSP '97*, Munich, Germany, 1997, pp. 1143–1146.

[14] B. Carpenter and J. Chu-Carroll, "Natural language call routing: A robust self-organizing approach," in *Proc. ICSLP '98*, Sydney, Australia, 1998.

[15] D. Fournier and B. Crémilleux, "Using impurity and depth for decision tree pruning," in *Proceedings of the Second International ICSC Symposium on Engineering of Intelligent Systems*, Paisley, Scotland, 2000.