

A Software Agent for Detecting and Correcting Speech Recognition Errors Using a Knowledge Base

Luis R. Lopes, University of Cape Town

Abstract—Common speech recognizers based mainly on statistical methods, can produce recognition errors that result in sentences containing grammar or logical errors. We describe a software agent that processes the output of a speech recognition engine and detects syntactic or semantic errors. It also attempts to correct the recognition error, when the correct word is obvious in the context of the output sentence, using the information in a common sense knowledge base.

Index Terms—Artificial Intelligence, Natural Language Processing, Speech Recognition, Spoken Language Processing

I. INTRODUCTION

THE output of speech recognizers often contain sentences that violate language rules, or that don't make any sense. Even commercially available dictation software packages are known to produce errors that give result to sentences that break grammar rules, or to nonsensical phrases, making it obvious to the user that the software does not have an understanding of human language.

We describe a software agent that post processes the output sentence of a speech recognition engine, to check for syntactic and semantic errors. The software agent goes further to attempt to correct the errors found, using information from a knowledge base. If it cannot correct the error it provides an indication, so that the main application can take an appropriate course of action if so desired, such as requesting the user to repeat the sentence.

II. KNOWLEDGE SOURCES IN SPEECH RECOGNITION

A. Standard Approach

State of the art speech recognizers use acoustic models based on Hidden Markov Models (HMMs) trained with a large speech corpus, typically from speakers that are representative of the target user population. The acoustic model therefore, captures the phonetic properties of the vocabulary to be recognized. To incorporate some knowledge of the language, a large text corpus is used to train a statistical language model. This text corpus will usually contain sentences that are related to the application

in mind. For example, a speech recognition engine to be used for dictation of radiological reports would use a text corpus of relevant radiological reports.

The most popular language models are n -gram language models, which essentially consist of establishing the probability of a sequence of n words. Although very effective to a certain extent, this technique has the disadvantage of not incorporating any real language knowledge [1].

B. Using Context and Knowledge

There are many discussions in the literature, that cover the use of context, or knowledge sources in speech recognition, most of them applying statistical methods [2] [3]. Perhaps the work that most closely resembles the system presented here is that of Lieberman *et al.* They have used more than 700,000 facts collected from the OpenMind Common Sense project at MIT, to detect a nonsensical sentence generated by a speech recognition engine and replace the erroneous word with one that makes more sense [4]. In this paper, we propose a system which not only handles semantic errors, but is also able to detect and correct syntactic errors. Although modest by comparison, the system described next, shows how simple experiments in artificial intelligence and speech recognition can be performed, without requiring the resources of a vast common sense knowledge base.

III. SYSTEM DESCRIPTION

At the time of this writing, the software agent to be described, can only correctly process sentences with a limited syntax or sentence structure. The first type of sentences that can be processed are those that begin with a determiner (D) followed by a singular noun (N) word, and end with a verb (V) in the present or past tense. We will call them sentences of type DNV, and examples are "the sun shines", "a bird sings" or "this woman slept". The second rule is a determiner, followed by a singular noun, followed by a verb in the present or past tense, followed by a determiner, and finally, a singular noun. We will call them sentences of type DNVDN, and examples are "this car overtook that bus", "the heat melts the ice", or "a girl won the race". Any sentences that do not follow this structure or word sequence will be deemed by the software agent to have a syntax error even if they are valid English sentences. We purposely limited the grammar complexity of the system in order to focus the research on the usefulness of the concept. However, the system can be extended to include additional sentence structures with a few modifications.

The system will work correctly with any vocabulary that conforms to the given grammar rules. The vocabulary is to

be established by the user, but obviously it should only contain words in the dictionary of the speech recognition engine.

Table I shows the initial vocabulary used on the first experiment. A much larger vocabulary can be used, but this implies a larger knowledge base as well. When a specific vocabulary set is chosen, the user needs to enter all the facts related to those words in the knowledge base. This is equivalent to teaching the software agent what we know about the world represented by the words contained in the vocabulary set. The vocabulary and the knowledge base can easily be modified with a simple text editor.

TABLE I
TEST VOCABULARY SET

D	N	V
A	Girl, Woman, Man	Eats, Ate
The	Banana, Apple	Dances, Danced
This	Dog, Cat, Rat	Laughs, Laughed
That		Runs, Ran
		Loves, Loved

The software agent uses two main knowledge bases in its library: a validation knowledge base (VKB) and a common sense knowledge base (CSKB). The VKB contains all the valid facts that we want the software agent to know about the world represented by the words in its vocabulary set, taking into account the grammar constraints specified. For example, using the words on Table I, our VKB would typically include the following facts: “people eat fruit”, “animals eat animals” and “people laugh”. All generalizations need to be entered in the knowledge base as well, i.e. the software agent needs to know that by “people” we mean “girl”, “woman” and “man”, in the agent’s vocabulary set. The agent will use the information in the VKB to determine if a sentence makes sense. The CSKB contains statements that are typical of the agent’s world and will always be a logical subset of the VKB. The software agent uses the CSKB to attempt to correct syntactic or semantic errors in a sentence. To illustrate the difference between the two knowledge bases, the information “animals laugh” will be contained in the VKB because it is a valid and correct statement, but might not be included in the common sense knowledge base. It is up to the user to correctly “teach” the agent what is valid and what is common sense, by choosing carefully what goes into each knowledge base.

When a sentence is passed to the software agent it first checks the syntax. If a syntax error is found, the agent attempts to correct it, otherwise it proceeds to check the sentence for semantic errors. If the sentence doesn’t make sense, the agent attempts to correct it. If no semantic errors are found either, the agent indicates that the sentence has passed both tests.

Checking for syntax errors is a relatively simple matter of verifying the sequence of words. Checking if a sentence makes sense involves comparing the sentence against the knowledge contained in the VKB. Attempting to correct syntactic or semantic errors is more complex. The agent evaluates the different possibilities using the information in the CSKB. It ranks the various hypotheses and chooses the

one with the highest ranking. If there are two or more hypotheses with the same ranking, it concludes that the solutions are ambiguous and gives up. The ranking is done by assigning specific statements in the CSKB such as “cats eat rats” a higher value than general statements such as “animals eat animals”.

To test the software agent we developed a graphical interface application that shows the result of processing a sentence typed by the user. It showed that the agent was able to correct sentences like “the cat eats the that” (syntax error) to “the cat eats the rat” or “the dog laughs that man” (semantic error) to “the dog loves that man”. We also incorporated the software agent into the SPHINX-4 speech recognition system, and tested it with two applications. One can transcribe the utterances in a recorded file, showing the output of the speech recognition and the output of the agent. The other is a live speech recognition application that also shows the recognized and processed result for each utterance spoken on a microphone.

IV. CONCLUSION

It turned out that the software agent wasn’t able to correct most sentences after being integrated with SPHINX-4. This was mainly due to the fact that there were several deletion errors in the recognition results, and the system was designed mainly for substitution errors. However, the few instances that the agent was able to make a correction show that there are advantages to this approach. We plan to use a larger vocabulary set for the next experiments, containing phonetically similar words where substitution errors are more likely to occur, and observe if a noticeable reduction in word error rate can be achieved with the software agent.

REFERENCES

- [1] R. Rosenfeld, “Two decades of statistical language modeling: where do we go from here?” *Proceedings of the IEEE*, vol. 88, pp. 1270–1278, August 2000.
- [2] A. Sarma and D.D. Palmer “Context-based speech recognition error detection and correction” in *Proc. Of the HLT-NAACL Conference: Short Papers*, Boston, MA, 2004, pp. 85–88.
- [3] S.R. Young “The MINDS system: using context and dialog to enhance speech recognition” in *Proc. Of the workshop on Speech and Natural Language*, Philadelphia, Pennsylvania, 1989, pp. 131–136.
- [4] H. Lieberman, A. Faaborg, W. Daher, and J. Espinosa “How to wreck a nice beach you sing calm incense” in *Proc. Of the 10th International Conference on Intelligent User Interfacing*, New York, NY, 2005, pp. 278–280.

Luis R. Lopes is a member of the Speech Technology and Research (STAR) group and a second year MSc(Eng) student in the Electrical Engineering department, at the University of Cape Town.

Prior to being granted leave to undertake postgraduate studies, he held the position of Governance Manager for the Information Technology & Services (IT&S) department at BP in Angola. This was after having worked in two different roles in the same organization in Angola and in Scotland.