TASK MODELLING BY SENTENCE TEMPLATES

Ute Kilian, Klaus Bader
Daimler-Benz AG, Research and Technology
Wilhelm-Runge-Str. 11, D-89081 Ulm, Germany
kilian@dbag.ulm.DaimlerBenz.COM

ABSTRACT

Speech recognition applications always face the problem of changing vocabulary and functionality. The use of speech recognition systems will become more attractive if the system user is able to define or redefine the task himself in a suitable manner. Modelling a new task normally requires the experience of a human expert and a lot of time. Aditionally, the expert always has to be contacted if system changes become necessary. In this paper we present a fully operational system for continuous speech recognition with a powerful user interface. Most of the internal aspects of the speech recognition system are hidden. The task may be divided into different subtasks corresponding to dialogue states. Each subtask is defined by a set of expected user utterances based on sentence templates. This definition is automatically transformed into a lexicon and a language model used by the speech recognition system.

1. INTRODUCTION

Systems for analyzing all aspects of continuous speech from word recognition up to linguistic representation cover many aspects of spontaneous speech (see [1]) or [2]). But modelling a new task requires the experience of a human expert and a lot of time. And it is quite difficult to get a data basis for calculating a language model, since during the definition phase of a new task no user utterances are available.
The use of speech recognition systems will become more attractive if the system user himself is able to define and redefine a task in a suitable way. Normally the system user is not very experienced in speech recognition or linguistics. Thus, most internal aspects of the system are hidden.
Applications with a small-to-medium vocabulary often can be covered by a *command and control* structure requiring syntactical restrictions of the speech input. But these restricitions lead to a much higher recognition rate being the precondition for user acceptance of system performance. And the syntactical restrictions offer a way to let the user define and redefine the task himself.
The system presented here is called ***Lexicon Development System*** (LDS). It offers a powerful user interface that facilitates the task definition to a high degree.

2. TASK DEFINITION

In a run through a speech based application the expected vocabulary and the expected user utterances normally change from one dialogue state to another. If this knowledge is incorporated into the system by the activation of different vocabularies and language models (refering to the corresponding dialogue state) the recognition rate is increased (see [3]). Furthermore, if a new task is modelled, no user utterances for the training of a language model are available. The task definition by sentence templates aims at

1. dividing the task into subtasks
2. defining for each subtask the expected user utterances

An example is given for a quality control task. The following information is to be entered into the system:

- worker identification
- product identification
- quality control (location and kind of defects ...)
etc.

Each line in the above example corresponds to one subtask.

The next step is to define for each of these subtasks the expected user utterances. This will be performed based on sentence templates. We decided to use sentence templates since their definition does not require much knowledge about linguistics or formal grammars. Each template represents a set of various sentences based on the following sentence units:

- **single words**
- **lists** (word categories like colours, but also enumerations like ‘John Smith’, ‘David Miller’ etc.)
- **loops** (words that may be repeated, e.g. digits for the representation of phone numbers)

Each list and loop is given a unique name, so it may be reused in different sentence templates.

In Figure 1 an example of one sentence template containing just one list is shown. The expected user utterances of the subtask *worker identification* are defined by a list of names. By a double click on the list name the list items are shown.

Figure 1: Sentence Template and List Items

The items of a list or loop may easily be modified. Consistency in the whole task definition is guaranteed by the use of unique names in the sentence templates. Furthermore, sentence units may be assigned an optional flag, i.e. these units may be used or omitted.

For each subtask a set of sentence templates is defined, see Figure 2. Each line represents one sentence template defining a set of various sentences. Each allowed sentence represents a complete parse of the sentence template (from the left to the right), optional parts may be used or omitted, the sentence units are treated like this:

- **single words** ⇒ just take it
- **lists** ⇒ choose one line of the list
- **loops** ⇒ choose one line of the loop, repeat that N-times

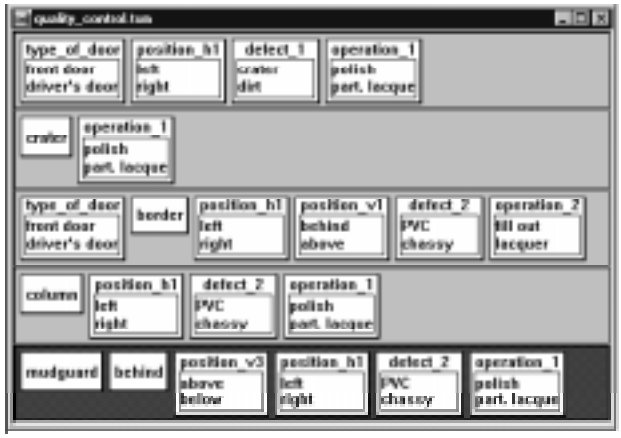

Figure 2: Sentence Templates for a Subtask in Quality Control

For each list and loop the first two items are shown in the sentence templates. The complete list is displayed by a double click on the selected sentence unit (Figure 1). Considering the first sentence template in this example allowed sentences are:

‘front door left crater partially lacquer’
‘driver’s door right dirt polish’
etc.

A special user interface provides the possibiliy to modify the sentence units and to modify their ordering in the sentence. A whole sentence template can be copied and moved to other subtasks.

3. MODULARIZATION

List and loops like *weekdays* and *daytimes* or *numbers* in a certain range are for general use and can be stored in a system template library. Creating a new subtask one can choose these sentence units for reuse.

In order to facilitate the subtask modularization LDS allows the use of different subtask modules in parallel. E.g. all expected user utternances concerning correction phrases are stored in a own subtask module. They can be used whenever correction phrases are expected.

Every task consists of at least one or more subtask modules. The selection of one or more subtasks in parallel is under control of the dialogue application. Different vocabularies and the corresponding language models can be activated. This results in an significant improvement of the recognition rate.

4. GENERATION OF THE LEXICON AND THE LANGUAGE MODEL

Based on the task definition a language model will be calculated consisting of one sublanguage model for each subtask definition. The language model is a syntactical bigram (SynBi, see [4]), i.e. the complete syntactical information given by the subtask definition is stored in the the form of a bigram. The SynBi will be directly integrated into the recognition process.

The SynBi calculation is performed in two steps. First, a graph is constructed based on the sentence units of the subtask definition. Herefore all sentence templates of one sublanguage model are inserted in the graph one after the other. Parts of sentence templates already contained in the graph are reused if this is possible (overgeneration is not allowed). Figure 3 presents this graph for the first two sentence templates shown in Figure 2.

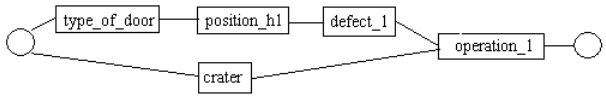

Figure 3: Graph Based on Sentence Units

In the second step the graph based on sentence units is transformed into a graph based on words. These words directly refer to those in the recognition lexicon. With the SynBi technique not only single words, but also word categories may be used for the language model description. Thus, the sentence units are substituted like this:

⇒ **single words:** just use it
⇒ **lists and loops:**
- *one word per line*: create a word category
- *multiple words per line*: create a subgraph

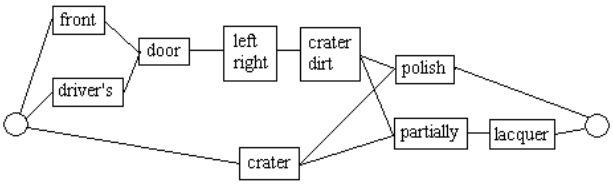

Figure 4: Graph Based on Words

E.g. the sentence unit *type_of_door* in Figure 3 contains two items: *front door* and *driver's door.* Both items consist of two words. Therefore a subgraph for this sentence unit is created (Figure 4). The word *door* is inserted in the subgraph during the insertion of *front door* and reused inserting *driver's door.*

During the creation of the word graph each word is given a unique index. I.e. a word occuring the first time is given the index 0, occuring the second time it is given the index 1, and so on. Based on the indexed words the SynBi technique integrates exactly the language defined in the sentence templates in the continuous speech recognizer working with a conventional bigram language model (for further details see [4]). I.e. the word graph is stored in the form of a SynBi (bigram with indexed words). The transition probabilities have predefined values suitable for most applications. But they may also be tuned by parameters.

Each indexed word of SynBi refers to one lexicon entry. I.e. there are more lexicon entries as in a conventional lexicon (caused by word indexing). Our lexicon is represented in a tree structure, i.e. only the word indices are multiplied, not the chains of HMM's. So there is no significant increase in lexicon size with an increasing number of word indices and no notable delay in the recognition process. The implementation of the recognition process itself remains unchanged.

The vocabulary for the whole task is defined merging all subtask vocabularies. Each word is assigned a flag indicating its subvocabulary membership.

The final lexicon generation is devided in three phases:

⇒ transcribing of the vocabulary words
⇒ generation of the HMM series
⇒ writing of the lexicon

The extracted vocabulary list contains just the orthographic representation of the words. However, for HMM-generation LDS needs the phonetic description. LDS uses a large phonetic dictionary to transcribe all the words. If a word has various phonetic descriptions the user is prompted a dialog to choose the right ones.
If LDS can't find a word all similar words are presented to the user to choose one as a boiler plate for the missing description. If an external phonetic dictionary book is used LDS provides an IPA-(International Phonetic Alphabeth) keyboard for interacting.

LDS transforms the phonetic description in a series of HMM units. This task is performed by a special rule interpreter which converts the phonetic description by applying the rules line by line. The set of rules corresponds to the current language and to the kind of speech recognizer in use.

The final procedure in lexicon generation is to write all the information of the vocabulary in a proper way so the speech recognizer can retrieve the vocabulary rapidly. Besides the orthographic representation the phonetic description and the HMM series mentioned above, a lexicon contains additional information concerning sub-vocabulary membership. Thus, the dialogue application can activate or deactivate parts of the entire vocubulary (in dependence of the current dialogue state) to improve the recognition performance.

5. FURTHER SYSTEM FEATURES

The system is running under Windows 95 and Windows NT. To support a large range of system platforms LDS stores both ASCII and binary information in a system independent manner. Thus, recognizers running on different processor architectures and operation systems will understand the output generated by LDS.

The system user may choose between a user interface in German or English. Online help functions are also offered in German and English. At the moment speech recognizers are available for American English, British English, German, French, Italian and Spanish.

Most internal aspects of LDS are hidden. As soon as the system user has defined a task and all its subtasks, the button *generate* will cause the generation of the lexicon and the language model. The system user does not need to know much about the internal system aspects described in chapter 4. Only if the phonetic transcription of a word is not found in the large phonetic dictionary the user is prompted to enter it.
But there are many possibilities for the experienced system user to tune the system changing predefined parameter values.

Another feature concerns garbage models. Pauses are automatically included in the SynBi, the garbage models (hesitations etc.) may be used in a predefined way or may be selected explicitly for the application.
The SynBi transition probabilities concerning the garbage models have predefined values suited for most applications. But again (like all other transition probabilities) they may be tuned by parameters.
This mechanism also permits the definition of wordspotting applications either based on words or on syllables (see [5]).

6. CONCLUSIONS

The use of speech recognition systems will become more attractive if the system user is able to define or redefine a task in s suitable way. In this paper we presented LDS, a fully operational system for continuous speech recognition with a powerful user interface. Most internal aspects of the speech recognition system are hidden. A task is divided into subtask. For each subtask sentence templates are defined describing the expected user utterances. The task definition is automatically transformed into a lexicon and language model used by the speech recognition system.

The user interface and predefined values for all system parameters enable a system user (even if unexperienced in speech recognition and linguistics) to define a working application. So the user gets familiar with speech technology. As soon as more knowledge on speech recognition or linguistics is achieved, LDS offers many possibilities for optimization.

REFERENCES

[1] A.Brietzmann, F.Class, U.Ehrlich, P.Heisterkamp, A.Kaltenmeier, K.Mecklenburg, P.Regel-Brietzmann, G. Hanrieder, W. Hiltl, ‘Robust Speech Understanding’, Proc. ICSLP 94, Yokohama 1994, pp. 967-970

[2] U. Ehrlich, G. Hanrieder, L. Hitzenberger, P. Heisterkamp, K. Mecklenburg, P. Regel-Brietzmann, ‘ACCeSS - Automated Call Center Through Speech Understanding System’, these proceedings

[3] C. Popovici, P. Baggia, ‘Specialized Language Models Using Dialogue Predictions’, Proc. ICASSP ‘97, pp.815 - 818, Munich 1997.

[4] U.Kilian, F.Class, A. Kaltenmeier, P.Regel-Brietzmann, ‘Representation of a Finite State Grammar as Bigram Language Model for Continuous Speech Recognition’, Proc. Eurospeech 95, pp. 1241-1244, Madrid 1995.

[5] H. Klemm, F. Class, U. Kilian, ‘Word and Phrase Spotting with Syllable-Based Garbage Modelling’, Proc. Eurospeech 95, pp. 2157-2160, Madrid 1995