DESIGN FOR A SPEECH-TO-SPEECH TRANSLATOR FOR FIELD USE

David Stallard, Premkumar Natarajan, Mohammed Noamany, Richard Schwartz, John Makhoul

BBN Technologies, Verizon
(stallard, pnatarajan, mnoamany, schwartz, makhoul)@bbn.com

ABSTRACT

We present the design of a proposed Two-Way Translator that we are developing for simple communication between US military or humanitarian personnel and non-English speakers in a foreign country. The translator uses a system-directed dialogue technique, and is intended for use in a handheld device. A novel aspect of our work is its use of statistical information extraction techniques to assist in the translation.

1. INTRODUCTION

Military and humanitarian personnel often need to communicate with residents of a host country who do not speak English. In a crisis situation, there will be little time to train personnel in the host country language, and human interpreters will often be in short supply. Portable devices for speech-to-speech language translation would therefore be very useful in such environments.

A partial solution to these problems is the recently developed DARPA One-Way translator [1]. This device consists of a PDA with a microphone. The English speaker, whom we shall call the operator, uses the one-way device by speaking into it one of a fixed set of utterances in English. The device uses a speech recognizer to determine which of the utterances he has spoken, and then uses table lookup to find a pre-recorded audio file containing the translation of that utterance into the language of the other party, whom we shall call the subject.

The DARPA One Way avoids having to translate in the reverse direction by designing all questions so that the subject can reply to them non-verbally. Examples are yes/no questions, which can be replied to by a gesture such as nodding or shaking the head, how-many questions which can be answered by holding up the corresponding number of fingers, and so on. WH questions with a finite set of answers, such as “What is your relationship to this person?”, are handled by asking a series of yes/no questions until the correct alternative is found.

This has proven useful in field trials, but has obvious limitations in terms of efficiency and expressive power. Our goal in the present work is to overcome these limitations by building a two-way translator, in which the subject can answer questions verbally in his own language, and the answers can be rendered into spoken or textual English for the operator.

Special challenges are presented by such an application. First, for many languages of the developing world, little or no speech recognition work has been done, and large corpora of transcribed speech data often do not exist. (Indeed, for some languages of current interest, such as Pashto, there is not even a standard accepted written form!) There are also substantial differences between dialects in many of these languages. For example, in Arabic, there are different colloquial versions in each country, all of them quite different from the formal version used in broadcasts. Additional factors complicating recognition of both subject and operator speech will include outdoor interviewing conditions, background noise, and operator and subject stress.

Because of these difficulties, a more restricted and limited approach seems appropriate, at least initially. Such an approach is analogous to that seen in many of the system-directed telephone dialogue systems that are in use today, where the problem of recognizing and understanding the user’s speech is kept to a minimum because the interaction is largely controlled by the system. In our system, it is the operator who controls the interaction by asking the questions, which in turn provides the context for interpreting the answers.

In spite of these attempts to limit the interaction, however, we can still expect speech recognition error to be an issue. Furthermore, subjects, especially those from different cultural backgrounds, will not always conform exactly to the instructions the system gives them, but may include additional words in their reply beyond just the information sought. For these reasons, the system must be able to extract the desired information from the response, while being robust to both recognition error and subject variation.

A novel aspect of our work is that we employ statistical information extraction techniques such as soft matching and topic classification to deal with these problems. These techniques (some of which have been proven in deployed spoken dialogue systems) are more robust to speech errors than rule-based approaches, as well as to lexical variations in the way different subjects express the same information.
To our knowledge, this has not been applied before to the speech-to-speech translation problem. Because understanding and information extraction are performed in the subject’s language, no direct language translation needs to be performed.

In the remainder of the paper, we present the design and overall approach of our system, with special emphasis on the information extraction problem.

2. APPROACH AND ARCHITECTURE

A block diagram of the system is shown in Figure 1. The operator’s speech is captured by a microphone and sent to an English-language recognizer, which produces a text version of his question, such as “What is your date of birth?”. The resulting English string is then passed to the subject-language-rendering module, responsible for producing subject-language audio corresponding to its translation in the subject’s language. In the initial version of this system, following the DARPA One Way, this will be a simple table lookup module, which maps the English string to a prerecorded utterance in the subject’s language. (Later, this may be extended to allow greater flexibility in operator questions.) The prerecorded utterance is then played out to the subject.

The subject replies into the microphone, and his speech is sent to a foreign-language recognizer, which produces a text string in the subject’s language. This text string is then passed to an information extraction module, which computes a frame interlingua representation of the relevant meaning. A frame consists of a type and set of slot/filler pairs. An example is the following, which represents the date answer “June 22, 1982”:

```
[DATE month: June
day: 22
year: 1982]
```

The frame is then passed to an English-language text generator, which computes an English text string from it, using straightforward template-based generation. This string is then displayed to the operator on a small screen. Alternatively, it may be rendered into English speech by a speech synthesizer.

The crucial component of this system is the information extractor, which we describe in the remainder of the paper.

3. INFORMATION EXTRACTION

3.1 Overview

We classify the questions the operator will be able to ask the subject into three types, based upon the type of answer they elicit. The first type are closed-set questions, such as “What city did you leave from?”, or “What is your occupation?”, where the answer will come from some fixed set of names or other simple expressions. Simple yes/no questions can also be viewed in this way.

The second category we call structured-value questions, such as “What day did you leave?”, or “How old is this child?”, in which the answer is a more complex expression, such as a date, a time, or an amount. Structured values have constituent parts, such as the month, day, and year for a date, or the quantity and unit for an amount, which must be determined as part of the extraction process.

The third and most advanced category are open-ended questions, such as “Why did you leave your village?”, which will tend to elicit long narrative responses. The goal of information extraction here will be to classify the narrative into a sub-type, such as “insufficient food”, “fear of attack”, and so on.

We discuss each of these question types below.

3.2 Closed-Set Answers

Closed-set answers are names or name-like expressions that denote individual members of some set. Examples are the names of cities, companies, family relationships, occupations, and so on. Even the Boolean values corresponding to ‘yes’ and ‘no’ can be thought of constituting such a set.

Responses to closed-set questions can consist of a single word or a sequence of many of them. The problem of recognizing which answer was meant thus consists of mapping the response to the correct canonical form in the answer set. The challenge lies in the number of different ways a subject can express this canonical answer, and the possibility that the subject may generate a new form that the system has not seen before.

The following example, which is drawn from a deployed automatic directory assistance application, may give some idea of the problem. Consider the following, which are all different ways that users chose to request the number of the Department of Motor Vehicles:
• Department of Motor Vehicles
• DMV
• Registry
• I’d like the number for the DMV, please
• The number for the Registry of Motor Vehicles on Main Street

Of these forms, only the first one or two may appear in the directory.

To cope with this problem, we follow the approach taken in [2], which is based on a statistical matching paradigm. We seek the most probable answer given the utterance, or $A'$:

$$A' = \arg \max_{A_k} P(A_k | Ut)$$

Applying Bayes’ rule, we have:

$$P(A_k | Ut) = \frac{P(Ut | A_k)P(A_k)}{P(Ut)}$$

The probability of the utterance $P(Ut)$ is the same for all $A_k$, so it can be ignored. To compute $P(Ut | A_k)$, we use a two-state HMM as outlined in Natarajan (2002). One state corresponds to the utterance word being produced by a particular $A_k$ and the other to it being generated by a non-answer phrase (NA). The formula for computing $P(Ut | A_k)$ is:

$$P(Ut | A_k) = \prod_{q \in Sent} (a_0P(q | NA) + a_1P(q | A_k))$$

where $a_0$ and $a_1$ are the transition probabilities.

The advantage of such a model over regular pattern matching approaches (such as the longest-match heuristic) is its robustness to every type of variation, including variation in order of constituent tokens, deletion of tokens, and insertion of tokens that were not part of the original pattern.

Since the approach is language independent, it can be easily applied to any spoken language. Large amounts of training data are not needed; all that is required for an initial capability is one or two ways to say each value in the foreign language, and these can be provided by an informant for the language. When actual training data becomes available, it can be used to improve the performance of the system. The algorithm also assigns confidences to the choices it makes, so that if the system is uncertain as to whether it understood the subject, the operator can ask the subject for an explicit yes/no confirmation of the answer.

### 3.2 Structured-Value Answers

Structured-value answers are those with syntactic and semantic structure, such as dates, times, and amounts, and are given in response to queries such as “What date did you leave your city?” or “How old is this child?” Languages typically have simple grammars for expressing such values. Given this simple structure, there is no need for a translation component from the foreign language to English. Rather, a simple semantic grammar can be written for each type of value, and used to parse the resulting recognition string in order to extract the components.

Our approach is to write a semantic grammar for each structured value type, and attach semantic rules to the grammar rules. The following is an example which generates the Arabic date “الثاني والعشرين من يونيه عام ألفين واثنين” (“Twenty second of June, 2002”):

```
DATE => DAY_ORD من MONTH عام NUMBER
```

```
/  [date month: @3
day: @1
year: @5]
```

Each rule in the grammar is associated with a semantic rule (separated by a “/”) that specifies the interlingua meaning of the generated phrase as a in terms of the meanings of its parts. Meanings will either be frames, as in the example above, or terminals, such as numbers (“2002”) or symbols (“June”).

A bottom-up, all-paths parser is run on the input and produces as output a filled-in chart. Each constituent (completed edge) of the chart will have an associated meaning in the interlingua. To be successful, it is not required that the chart contain a constituent which spans the entire input string. Rather, the system extracts from the chart the longest parsed constituent that has a meaning of the desired type, which may be only a substring of the input. The meaning of this constituent is returned as the result.

Some inputs may result in a chart that has no constituent at all with the desired meaning-type. This may come about because of a shortcoming in grammar coverage, or simply because the subject spoke strangely: “We left our homes on the twenty second day of the month of June in the year of our Lord two thousand and two”. A fragment combination component is included to enable the system to understand such utterances.

This fragment combiner searches for constituents in the chart which can fill a slot in the target frame-type. A greedy algorithm enumerates them in order of length. For our example, the constituents would be:

```
YEAR =>* two thousand and two / 2002
ORDINAL_DAY =>* twenty second / 22
MONTH =>* June / JUNE
```

Given these slot fillers, generating a DATE frame is straightforward.

In some cases, the fragment combiner may only find enough filler constituents to produce a partially filled-in frame. This frame may be still be useful, however, if it contains enough information to be rendered in English. For example, the frame:
Given an existing corpus of refugee narratives, in any language, we can potentially use it to classify similar narratives in any other language. All that is needed is a bilingual dictionary between the languages, generated from parallel corpora or other sources. The narratives for the seed language could be hand translated to English on a one-time basis, to make parallel corpora easier to obtain. Likely sources of parallel corpora include BBC Radio News, which has services for Pashto and other languages, and the Voice of America.

4. CURRENT STATUS

As an initial language to work with, we have chosen Modern Standard Arabic. The bottom-up chart parser and interpreter described above has been implemented in C++, along with a structured-value grammar for dates, times, and amounts in Arabic.

We are currently in the process of collecting closed-set answer data from native Arabic speakers, using scenarios drawn from the refugee domain, and expect to have a prototype system in the next few months.

5. ACKNOWLEDGEMENTS

This work was sponsored, in part, by DARPA and monitored by SPAWAR Systems Center under Contract No. N66001-99-D-8615.

6. REFERENCES


