Different Approaches to Build Multilingual Conversational Systems

Marion Mast, Thomas Ross, Henrik Schulz
IBM European Speech Research
Vangerowstr. 18
D-69115 Heidelberg
Henriks@de.ibm.com

Heli Harrikari
NOKIA Research Center
Itämerenkatu 11-13
FIN-00045 NOKIA GROUP
heli.harrikari@nokia.com

ABSTRACT

The paper describes developments and results of the work being carried out during the European research project CATCH-2004 (Converse in AThens Cologne and Helsinki)[1]. The objective of the project is multi-modal, multi-lingual conversational access to information systems. This paper concentrates on issues of the multilingual telephony-based speech and natural language understanding components.

1. INTRODUCTION

CATCH-2004 aims to develop a multilingual, conversational system providing access to multiple applications and sources of information. The system is designed to support multiple client devices such as kiosks, telephones and smart wireless devices. It will also allow users to interact with multiple input modalities. The architecture is composed of two major frameworks: a server-side Multi-Modal Portal providing a flexible middleware technology for interacting with multiple clients in multiple modalities and languages, and a telephony-based conversational Natural Language Understanding (NLU) system.

The common core application in CATCH-2004 is so-called 'City Event Information' (CEI) that provides information about cultural events in the three cities involved in the project. This paper however concentrates on the further developments after CEI, namely 'Sports Application' (SPA) developed for English, German, and Greek as well as 'Program Guide Information Service' (PGIS), which is available in English and Finnish. The SPA application is able to answer requests about sport events taking place during the upcoming Olympic Games in Athens in 2004, including also additional information such as olympic and world records, the history of the sports and the venues. PGIS is an electronic program guide where users can obtain information about TV programs based on various search parameters, such as channel, date, program type, and perfomer. Additional information about programs is also available, for example, description, restrictions, and duration.

[1] The project is co-funded by the European Union in the scope of the IST programme (IST 1999-11103). The paper represents the view of the authors.

This paper discusses architectural aspects of the telephony-based multilingual conversational NLU system. In Section 2, different approaches to build multilingual conversational systems and specific multilingual aspects of the components are discussed. Section 3 gives more details about the specific components. The initial results comparing the performance of different architectures are shown in Section 4. Finally we present some ideas for further work.

2. APPROACHES TO BUILD MULTI-LINGUAL CONVERSATIONAL SYSTEMS

2.1 General architecture

Let us begin by demonstrating the overall system architecture. This is shown in Figure 1.

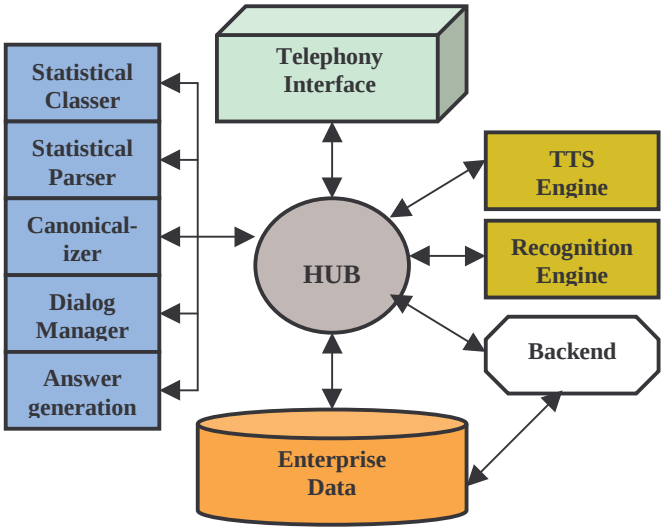

Figure 1: General system architecture

Communication between the components is executed via a hub. The hub works as a dispatcher that calls and routes information between involved modules. The telephony interface handles basic telephony functions, such as accepting and disconnecting calls, detection of hang-ups, recording and playing back audio material, and DTMF tone detection. After recording an utterance, the speech recognition module is invoked. The decoded text is

delivered to the statistical classer, where simple application-specific concepts are identified. The canonicalizer then extracts canonical values for these basic concepts, followed by the statistical parser that computes the semantic parse from the classed sentence. The dialog manager interprets the parser result in the dialog context, requests backend information, and produces the system reaction for the user utterance, which is then passed to the TTS engine.

2.2 Multilingual architecture

Next we will describe different approaches of building a multilingual conversational system. Particular attention will be paid to the two architectural solutions, which have been implemented in CATCH-2004.

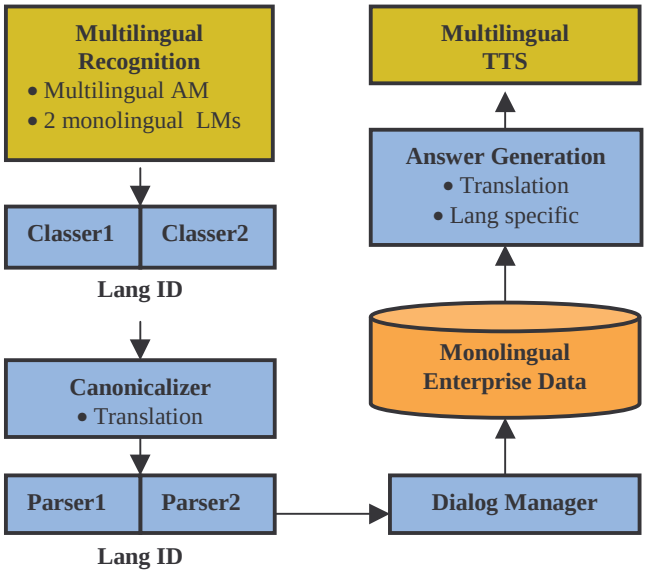

Figure 2: Multilingual architecture with the separation model

The scale of multilinguality for a conversational system can reach from parallel monolingual systems towards completely multilingual systems. In the former case the user selects the language in the very first utterance that will then be used during the entire conversation, whereas in the latter case, the system is based on single multilingual components enabling the language switch at any point of the dialog.

A system built from separate monolingual modules must decide at the beginning of each conversation, which is the language that the user prefers. This might be achieved in different ways. The user could be asked to select the preferred language by using touch tones, or alternatively a language identification module can be utilized where the first utterance determines the language to be used. The advantage of this approach is that a multilingual system can be constructed with minimal effort assuming that the monolingual systems for different languages already exist. The drawback is however that not every telephone is equipped with touch tone capabilities. Naturally, the approach with language identification is more convenient for the user, unless the determination of the language fails and the user will be stuck with a language (s)he does not understand.

Our first architectural solution for a multilingual system is a step forward from the situation just described. Figure 2. illustrates the structural properties of the system. Only relevant components are shown here (cf. Figure 1). We demonstrate the architecture with two languages only.

Figure 2 demonstrates how the two parallel monolingual modules are maintained for the classer and parser in the NLU, and similarly for language modeling (LM) and vocabularies in the speech recognition. Despite of running two parallel monolingual components simultaneously, the user is not bound to one language for the rest of the conversation. By maintaining all modules for all languages continuously active, the language can be determined separately for each utterance. The switch of the language is possible at any stage of the conversation. Language identification of the current utterance is executed either after the classer or parser. Both monolingual classers provide the preferred output for the utterance, after which confidence measures are compared, and consequently the language of the utterance is determined. Alternatively, the language might be determined by comparing both monolingual classers and parsers, and then selecting the best scored parses. Furthermore, if a monolingual backend database is given, a translation mechanism can been implemented. First, the translation executed in the canonicalizer guarantees that the relevant parts of the database query will be in the correct language, whereas the translation in the answer generation leads to the answer in the same language as the original query was uttered. The advantage of this approach is that the maintenance of the system is considerably easier than for systems with fully multilingual components: incorporating various monolingual components into the system requires only slight modifications. Furthermore, having separate modules for separate languages might also result in better accuracy (see Section 4 for evaluation).

The next possible approach is similar to our approach in Figure 2., but instead of maintaining separate decoding paths for the different languages in LMs, a fully multilingual speech recognition can be combined with many parallel monolingual NLU and the language-independent dialog system. In this case the language must be identified during or after the speech recognition. Again this might be done in various ways. First, the speech recognizer can deliver the decoded utterance together with a language ID, or a separate language identification module might be built to work on the decoded utterance. The third possibility would be to have the NLU components to determine the language in question. Drawback of this approach is probably less accurate speech recognition than with monolingual systems, and the possibility of incorrect language identification.

The last approach is to build a conversational system where all components are fully multilingual. This is the second approach that we have implemented in our project. Figure 3 demonstrates the structure of the relevant components. Fully multilingual speech recognition, NLU and dialog manager are employed. NLU continues to determine the language here as well, but now we have implemented two alternative ways of language identification.

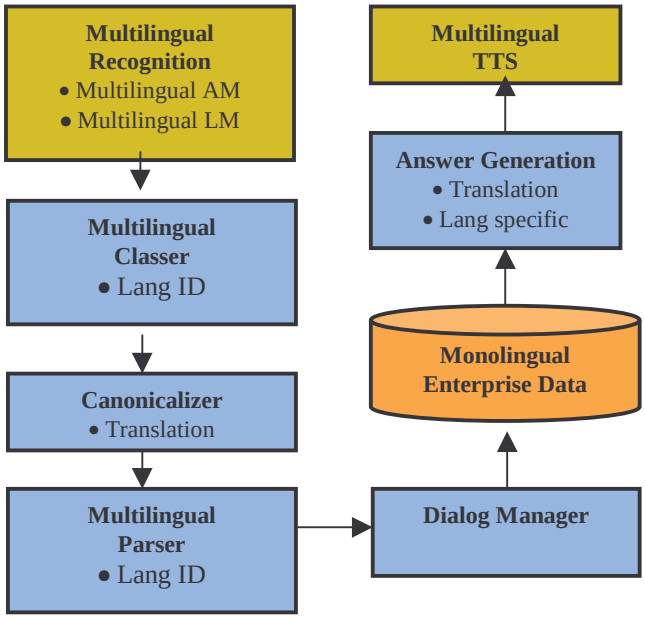

Figure 3: Multilingual architecture with multilingual modules

The decision can be made by the classer based on specific language tags attached to the sentences, or alternatively during the parsing with similar tags. This architectural solution allows the language switch between utterances (similar to the approach in Figure 2) and even within an utterance.

The following aspects are to be born in mind when deciding for one approach or the other. Accuracy issues could be an argument against fully multilingual components. Maintenance issues also support a solution where at least certain components are separate for different languages. The same applies to adding new languages into the system: as long as the monolingual application is available, incorporating its components into the multilingual application is relatively easy.

Another aspect is the environment in which such a system will be deployed. In the case of a telephony system, which provides communication in the users' mother tongue, it's rather unlikely that a user will switch between languages in an utterance or even between utterances. When deploying a kiosk system (e.g. at a train station or airport), the demands might be quite different. For example, it may be difficult for the system to determine the end of a dialog with one user, and the beginning of the next one. Thus, a system that allows switching of languages between utterances might be preferable.

The final issue is the backend data. Either the data exist for all required languages, or in one language only, and translation is employed. This is the situation with e.g. the PGIS system. Generally, as machine translation is not a solved problem, having to deal with translation may introduce further problems.

3. Components

We will next turn to a more detailed discussion about the various components. Particular attention will be paid to the requirements set by the various multilingual approaches.

3.1 Acoustic model

The speech recognition engine used here is the IBM ViaVoice decoder. It is described in some detail e.g. in [1]. In this section, a short summary of basic aspects of the training procedure as well as a brief overview of the recognition system are given.

The training of the 8kHz system is a bootstrap procedure based on an initial acoustic model built from downsampled 22kHz data only. In a first step, cepstral features and their first and second order derivatives are computed. With the bootstrap system, the training data is viterbi aligned against its transcription. Based on this alignment, a decision tree is constructed for subphonetic HMMs by querying phone context. The data corresponding to a subphone (leaf) is clustered and consequently modeled by a mixture of Gaussians with diagonal covariance matrices. The so created models are refined by running a few iterations of the forward-backward training algorithm, see e.g. [4].

Training data consists of both real telephony data (landline, cordless, cellular phone) and downsampled 22kHz office correspondence data with the latter having a higher percentage of rich context utterances. The telephony data set comprises utterances from various domains ranging from digit strings, numbers, spellings, office correspondence to spontaneous utterances. Using additional downsampled 22kHz data has shown better results than training with real telephony data only. In particular, this is true if high quality ISDN data is present in the test set.

The main purpose for a multilingual acoustic model in this scenario is to enable decoding of utterances from more than one language. But benefits can also be expected when decoding monolingual data: decoding non native speakers as well as decoding words foreign to the respective language might profit from a multilingual acoustic model.

An important step in building a multilingual acoustic model is the definition of a phonology common to the languages covered. All models compared here are based on a phonology described in [6] (CPA-3). The main difference between models considered here is the number of utterances per language in the training set. Other parameters, e.g. the number of HMM states or Gaussian mixture components are similar but not identical due to stochastic aspects in the training procedure. In the initial lexeme selection step, spellings are tagged with language specific identifiers to avoid erroneous cross language selections.

The four acoustic models compared are different with respect to the training data as follows:

- m_{uk}: UK English data only
- m_{gr}: German data only
- m_1: English (UK and US), German, Finnish: 30k; Italian, Spanish, French: 10k; Greek: 3k

- m_2: additional 20k for Spanish, French and Italian and additional 60k for German

other parameters, e.g. number of HMM states, number of Gaussians, were within the same range. The selection of model m_1 is based on positive results on tests against non native English test data, model m_2 was selected to have comparable amounts of German and English in the training.

3.2 Language model

Here, widely used 3-gram language models are used, both word- and class-based. Given that conversational NLU systems are based on relatively small training corpora, class-based LMs show advantages [2]. During the recognition each speech frame is labeled and passed to the acoustic fast match which uses continuous density Hidden-Markov-Models (HMM). For all words with a high fast match score (the fast match list) a LM score is computed based on the sequences of words decoded so far. This reduces the number of words for which the computationally more expensive so-called detailed match has to be computed in the next processing step. A heuristic search algorithm determines at each step, which paths to expand. The best path covering all input data is selected as the decoder output [1].

For constructing a multilingual LM two potential methodologies exist. In the first approach, statistics are computed for all corpora separately, as is the case in our first approach (Figure 2). In the second approach, the corpora for each language are merged and statistics as well as the vocabulary are based on the merged corpus (Figure 3). When using a separate vocabulary and language model for each language, an utterance is decoded within the respective LM between two firm up points. A firm up point is determined by a number of events, e.g. the end of an utterance, silence, or noise, and it also depends on optimization criteria for the decoding process. Once a firm up point has been determined, the best decoding path across the separate models is selected. This will favour language-consistent decoding. A common vocabulary and language model may suffer from unigram probabilities that occur for words with spellings in more than one language. These words are cross-points of the decoding paths for the respective languages and may lead to a language switch between the firm up points. For both Finnish and English, the PGIS application contains a multilanguage vocabulary (movie names, actors etc.). Apart from a potential overlap of unigram probabilities, this leads to a number of bi- and trigrams across these names that may enable languge switches between firm up points.

Furthermore, the idea of incrementally adding new languages to a system, whose framework is language independent, would benefit from the aspect of having separate vocabularies and language models. The incremental addition of new languages is a strong cost aspect for building such systems.

3.3 NLU

For NLU we use a two-level parsing strategy. On the first level, the **classer** identifies simple semantic concepts. Usually the assigned expressions of these concepts are used as parameters to set up a backend request. The classing is done with a statistical parser trained from a corpus of annotated sentences [7]. Initially, each word within a sentence has to be tagged with its appropriate class tag. Tagged words are combined to labelled constituents depending on whether they can be assigned to the same semantic concept.

On the second level, the **parser** extracts semantic concepts as well as the focus and intention of the utterance. It is also trained from annotated sentences, whereby the partial expressions assigned to simple semantic concepts identified by the classer are replaced by identifiers of the respective concept. The parser follows the same statistical parsing methodology as the classer, but more layers of labels may be necessary to assign detailed semantic concepts.

The tags and labels are used by the dialog manager to identify the current task and decide the next step. The tags and labels used for an application are the same across all languages. Only the values of the simple semantic concepts, e.g. dates, names or numbers, may differ between the languages. A detailed description of the monolingual conversational systems developed during CATCH2004 is provided in [3].

3.4 Dialog Management

The Form-based Dialog Manager (FDM) is a framework for free-flow dialog management [5]. It allows a task oriented, mixed initiative dialog with a user. The framework can handle various types of dialog features such as asking for missing information needed to perform a task, clarifying ambiguities, inheriting information from the dialog context, and switching to directed dialog if needed. Each task is modelled as a form that has the knowledge of the information needed to perform the task and how to perform the task, e.g. calling the backend, selecting the answer according to the reply from the backend. For each user utterance the canonicalized attribute-value pairs created from the class-tree and the attribute-value pairs extracted from the parse-tree are fed into the slots of the respective form. Furthermore, it scores the forms and selects the one most suitable for the information provided with these attribute-value pairs. Since the classer and parser use the same tags and labels across languages, the dialog manager is language-independent.

The FDM triggers system responses to the user by composing textual messages from templates. The selection of the appropriate template is dependent on the dialog situation. Usually a template is designed to concatenate valuable slots of the respective suitable form with pre-defined expressions. For the multilingual approach the answer templates are provided for each language and the system selects the template according to the language used in the actual utterance.

4. EVALUATION OF MULTILINGUAL COMPONENTS

4.1 Speech recognition

Table 2 and table 3 give recognition results on four different test sets: set T_{gr} is a sample of spontaneous German utterances taken from a running application with queries about the SPA

application. Set T_{en1} is a similar test set in English. While T_{gr} utterances are taken from native speakers only, T_{en1} contains approx. 45% UK English utterances, no native US English speaker and 55% utterances from non native English speakers. Set T_{fi} consists of Finnish utterances (native) taken from the PGIS application and set T_{en2} is the equivalent set for English PGIS dialogues. Here, we only have non native English speakers in the test set. This comparison shows both the influence of multilingual data in the training set as well as the effect of multilingual vocabularies and language models. As can be seen in Table 2, for both English and German, the models trained on monolingual data perform best. For both German and English, adding additional 60k German utterances to the training set has an expected effect: While the German word error rate is reduced by approx 10% relative (8%, if the multilingual task is used), the English word error rate goes up approx. 8% (both cases). Decoding against a multilingual LM shows for German less (relative) degradation than for English and Finnish.

	m_{uk}	m_{gr}	m_1	m_2
T_{gr} - mono. LM	-	18.2	24.4	22.0
T_{gr} - multi. LM	-	-	24.6	22.6
T_{en1} - mono. LM	17.1	-	20.2	21.9
T_{en1} - multi. LM	-	-	23.8	25.7

Table 2: Decoding results (word error rates) for English (T_{en1}) and German (T_{gr}) test sets against mono- and multilingual acoustic models and tasks.

For the PGIS test sets, a comparison of monolingual, parallel and combined multilingual LMs and vocabularies were conducted with acoustic model m_1. Note that all speakers in T_{en2} are non native. The results support initial considerations that the accuracy might profit from separated LMs

	monolingual	parallel (2 LMs)	combined (1 LM)
T_{en2}	19,2	27.0	29.7
T_{fi}	5.6	6.4	7.9
Total		16.7	18.8

Table 3: Comparison of decoding results (word error rates) for English (T_{en2}) and Finnish (T_{fi}) test sets against mono- and multilingual tasks.

4.2 NLU

The data we used to test the performance of the NLU components in the different architectures were collected during user tests with the English and German monolingual SPA telephony system. The German test set contains around 330 sentences, and the English one around 350 sentences. As mentioned above, the English test persons were mainly non native English speakers.

The data were tested with the respective monolingual classer and parser, and a multilingual classer and parser for German and English. The multilingual components were build by simply merging the monolingual training corpora without further tuning of the classer or parser. Additionally we evaluated the parallel approach by testing the data with both monolingual classers and parsers and choosing the best scored parse. Table 4 demonstrates the classer and parser accuracy for the three approaches. The accuracy is calculated on a sentence level, only if each tag and label for each word in the sentence is correct, the sentence is rated as correct. The number of errors which really influenced the system behaviour negatively was much smaller, e.g. for the German data 3 % for the classer and parser.

	Monolingual	Parallel	Multilingual
German Classer	96 %	96 %	93 %
English Classer	94 %	93 %	92 %
German Parser	83 %	83 %	82%
English Parser	82 %	81 %	82 %

Table 4. Classer and parser accuracy for English and German data with monolingual, parallel and multilingual approach

Of course the results represented are only initial tests and need to be confirmed by further tests e.g. with more languages and more test data. For the classer the accuracy for the parallel approach seems to work better than the multilingual one. On the parser side the numbers are close together.

5. CONCLUSION AND FUTURE WORK

In this paper we have presented different architectural approaches to multilingual conversational systems and also demonstrated the initial results from the component evaluation of the different solutions. In many respects, the results show the superiority of the multilingual architecture with parallel language-specific components. First of all, the evaluation demonstrates better or equal performance accuracy of the parallel system. Furthermore, the maintenance of this approach has turned out to be relatively easy: the system can be easily constructed from monolingual systems, and also adding new languages requires only slight modifications in the overall system. However, the parallel approach is more expensive when it comes to processing power for all parallel components.

Naturally, various issues related to the different architectural approaches require further investigation. First, the effects of the different architectures on the performance accuracy and speed must be looked into in more detail, and incorporating additional languages is also an issue in the near future. Second, the influence of non-native speakers on the performance of the system has to be taken into consideration more thoroughly than has been done so far. Furthermore, of all possible architectural variations, we have so far implemented only two solutions. It could be worth while to investigate the other architectures proposed in Section 2, and compare them with the ones presented in this paper. Finally, our current research has solely concentrated on the performance issues of the different multilingual approaches. However, usability issues also require attention from us developers in order to properly decide, which approach is the most suitable one for the various needs for multilingual conversational systems in the future.

6. REFERENCES

[1] P. Gopalakrishnan, D. Nahamoo, L. Bahl, P. de Souza, and M. Picheny. Context-Dependent Vector Quantization for Continuous Speech Recognition. Proc. of the IEEE Int. Conference on Acoustics, Speech, and Signalprocessing, Minneapolis, 1993.

[2] G. Maltese, P. Bravetti, H. Crépy, B. Grainger, M. Herzog, F. Palou. Combining Word and Class-based Language Models: a Comparative Study in Several Languages using Automatic and Manual Word-Clustering Techniques, EUROSPEECH 2001, Aalborg, Denmark, 2001.

[3] M. Mast, Th. Ross, H. Schulz, H. Harrikari, V. Demesticha, Y. Vamvakoulas, J. Stadermann: A Conversational Natural Language Understanding Information System for Multiple Languages, NLDB 2001, Madrid, Spain, 2001.

[4] F. Jelinek, Statistical Methods for Speech Recognition, Cambridge, Ma., The MIT Press, 1997.

[5] K. A. Papineni, S. Roukos, and R. T. Ward, Free-Flow Dialog Management Using Forms. Eurospeech 99, Budapest,

[6] F. Palou, P. Bravetti, O. Emam, V. Fischer, E. Janke: Towards a Common Phone Alphabet for Multilingual Speech Recognition, ICSLP 1999

[7] F. Jelinek, J. Lafferty, D. Magerman, R. Mercer, A. Ratnaparkhi and S. Roukos, “Decision Tree Parsing using a Hidden Derivational Model”, Proc. of the ARPA Human Language Technology Workshop, pp 272-27, 1994.