A Conversational Natural Language Understanding Information System for Multiple Languages

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

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

Vasiliki Demesticha, Lazaros Polymenakos,
Yannis Vamvakoulas
IBM Hellas
Kifisias 284
GR-15232 Chalandri
valia_demesticha@gr.ibm.com

Jan Stadermann
University of Duisburg
Lotharstr. 65
D-47048 Duisburg
stadermann@fb9-i.uni-duisburg.de

Abstract: The paper describes aspects of the development of a conversational natural language understanding system done during the first year of the European research project CATCH-2004 (Converse in AThens Cologne and Helsinki). The project is co-funded by the European Union in the scope of the IST programme[1].
Its objectives focus on multi-modal, multi-lingual conversational natural language access to information systems. The paper emphasises on architecture, and telephony-based speech and natural language understanding components as well as aspects of the implementation of a City Event Information system in English, Finnish, German and Greek. The City Event Information system accesses two different databases in Athens and Helsinki using a common retrieval interface. Furthermore the paper singles out methodologies involved for acoustic and language model of the speech recognition component, and parsing techniques and dialog modelling for the conversational natural language subsystem. For the implementation it outlines an incremental system refinement methodology to adapt the necessary system components to real-life data. It addresses the implementation of language specific characteristics of the language specific components and a common dialog design for all four languages. Finally, it presents further the prospects for further developments.
The paper represents the view of the respective authors.

[1] IST 1999-11103

1 Introduction

Information services are more and more available and are becoming increasingly complex. It has therefore become desirable to provide users with conversational access to information by means of human-like behaviour of the system. It is the intention to resolve user queries in a mixed initiative dialog by sharing applicable information of the system and leading the dialog to retrieve the desired information.
With several thin devices (such as telephones, smart wireless devices) that can potentially be used by anyone to access information, designing interfaces that enable any (expert or novice) user friendly and pervasive access to automated services information is becoming of paramount importance.
CATCH-2004 aims to develop a multilingual, conversational system providing access to multiple applications and sources of information. The system architecture is designed to support multiple client devices such as kiosks, telephones and smart wireless devices. It also will 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 [5].
The paper focuses on architectural and methodological aspects of the telephony based conversational NLU system, and describes the application developments been done for a City Event Information system (CEI) for two cities: Athens and Helsinki.

2 Architecture and components of a telephony based conversational natural language understanding system

In the following sections we describe architecture and components of the telephony based conversational natural language understanding system.

2.1 Telephony based Speech Components

Telephony Device Driver (TDD). The TDD acts as audio data interface between different telephony systems, and the Speech Recognition and Text-To-Speech component. It also detects telephony DTMF (Dial-Tone-Multi-Frequencies) signals, which may be used by the IVR Hub to control the conversational application. The IVR Hub also controls the entire call flow between the speech components and the conversational NLU subsystem. It passes the backend requests of the NLU subsystem to the backend interface.

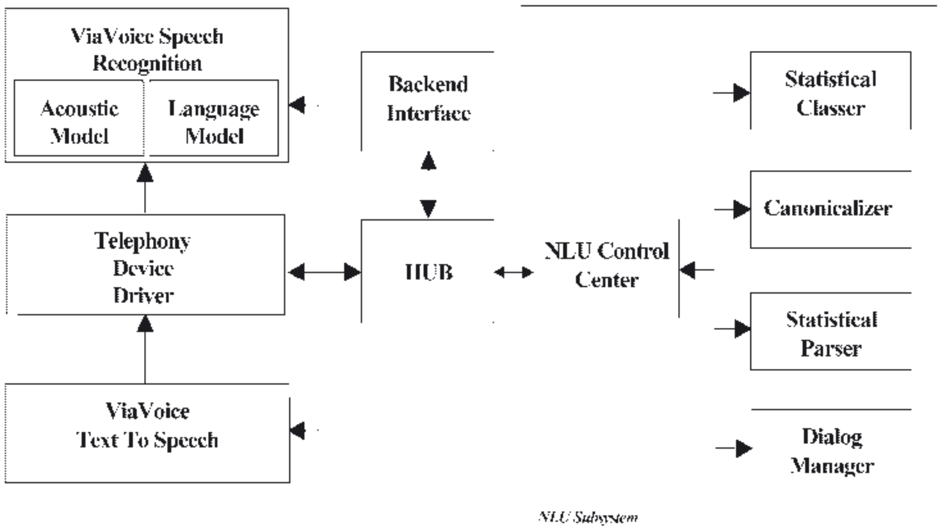

Fig. 2.1: Overall architecture of the telephony based conversational NLU system

Speech Recognition. The speech recognition component uses IBM ViaVoice. It receives telephony audio data from the telephony device driver, recognises the spoken utterances, and sends the transcribed utterances to the IVR Hub for further processing of the NLU subsystem. Regarding the speech recognition for conversational NLU systems, special emphasis needs to be put on the development of acoustic model (AM) and language model (LM).
The AM is created for 8kHz telephony-based speech recognition using an initial AM built from downsampled 22kHz data only. For further training the data consist of both real telephony data (landline, cordless, cellular phone) and bandsampled 22kHz office correspondence data. The telephony data set comprises utterances from various domains: digit strings, numbers, time, date, spellings, a limited amount of office correspondence and spontaneous utterances.
During the recognition each speech frame is labelled and passed to the acoustic fast match which uses continuous density 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 in the next processing step the computationally more expensive so-called detailed match has to be computed. A heuristic search algorithm determines at each step which paths to expand. The best path covering all input data is chosen as decoder output [4].
The LM consists of statistics that are used in predicting word sequences-which words are likely to follow one another. In general for this purpose, word-based trigram LM or class-based trigram models can be used. Given that conversational NLU systems are built on relatively small training corpora, class-based LMs are rather convenient.
Text-To-Speech (TTS). The TTS component uses IBM ViaVoice Outloud to generate a spoken response. It is a formant-based synthesizer and dynamically allows modifications of prosodic features like f0 and phone duration. The TTS triggered by the IVR HUB

creates the audio data for a given textual message, and sends those to the telephony device driver.

2.2 Conversational Natural Language Understanding Subsystem

The NLU subsystem is based on a two-level parsing (classer, parser) [2], a form-based dialog manager (FDM), a canonicalizer, a backend interface component, and central dispatcher calling and routing information between involved components.
Classer. 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. 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.

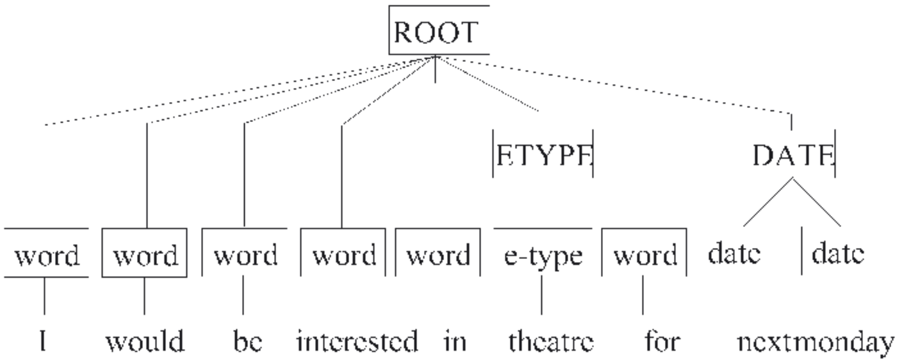

Fig. 2.2: Annotation scheme for classer parsing trees

During training the classer builds statistics for the following relations: words assigned to a tag, tags assigned to a label, labels assigned to a label of combined label constituents, and for labelling a node. The statistical model represents the characteristics of trained sentences that lead to certain tags, labels, or extensions.
When classing a sentence, the classer builds a parse tree from the words up. It frequently encounters more than one possible modification to the tree, each with a different probability. The classer maintains a set of partial parse-trees at any given time, each with a different probability. It searches for the most probable parse-tree. The classer returns the n-best scoring parse trees.
Parser. The parser extracts semantic concepts as well as focus and intention of the utterance. It is 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 even more layers of labels may be necessary to assign rather detailed semantic concepts.
Canonicalizer. The semantic concepts identified by the two-level parsing are also assigned to partial expressions of an utterance, those values are provided as parameters to the backend function. The canonicalizer transforms these partial expressions into a

format required by the respective backend function. For the transformation, a flat structure of attribute-value pairs is derived from the class-tree, and fed into the canonicalizer. Simple pattern matching techniques, word spotting algorithms or simple grammars are applied depending on the identified semantic concept and the complexity of the assigned partial expressions. For example, the partial expression 'next monday' is identified as concept 'date' and was canonicalized into '13/01/01'. Furthermore, the expression 'the national theatre of athens' was identified as a particular name and therefore harmonized into 'the athens national theatre'. Additional canonicalization can be performed as part of the backend application functionality.
Form-based Dialog Manager (FDM). The FDM is a framework for free-flow dialog management [1]. It allows a task oriented, mixed initiative dialog with a user. The framework can handle various kinds of dialog features like 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 knows about the information needed to perform the task and how to perform the task, e.g. calling the backend, choosing the answer according to the answer 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 slots of respective form. It furthermore scores which form is the most suitable for the information provided with these attribute-value pairs.
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. Flags may be used to indicate a special treatment for the TTS. Especially flags to force a temporary language switching for words not belonging to the main language improve the perceivability. The textual message is finally send via the IVR Hub to the TTS.
Backend Interface. The backend interface builds a bridge between NLU subsystem and backend application functionality. It receives requests of the FDM, calls the backend application, and passes the results back.

3 Development of a Conversational Multi-Language City Event Information System

As part of the developments a conversational city event information system (CEI) was implemented applying the architecture and components described above. CEI demonstrators are to be deployed for two cities: Athens, supporting English, German and Greek, and Helsinki supporting English and Finnish.
The CEI design considers the intention of people generally interested in some cultural events to retrieve information for a particular date an event is taking place, location and time. Requests may demand information about the published events and can be constraint by date, time, locations, type or event names. Depending on the expressed constraints, a list of matching events is passed back to the user. Furthermore the ongoing dialog narrows the scope of those events, that are of particular interest. Once an event is chosen to be of particular interest, user queries are referring to this particular event.

Apart from the general information on the event list the user can retrieve information for the address of the location where the event takes place, the prices of the tickets for the specific event, phone numbers of the locations, date and time of the event and information on the ease of access for the physically impaired people.
The development is split into two major phases. The first phase is used to build an initial bootstrap system from handcrafted sentences. The second phase will use the initial system to collect acoustic real life data and to test the dialog design. The acoustic real life data will be transcribed to improve the AM, the LM and the NLU subsystem. The refinement methodology can be used iteratively to reach a higher accuracy and enhance the understanding capabilities.
In the sequel we describe several aspects of the language specific CEI implementation based on the components described in the chapter above.

3.1 Speech Recognition

For the speech recognition component of the Athens CEI implementation monolingual acoustic model (AM) have been built respectively for English and German. The Helsinki CEI is build on a bilingual AM covering Finnish and English. Resources for the speech recognition components for Greek are currently in development.
Furthermore, application specific LM were developed for English, Finnish and German. The corpora for these LM originate from the initial data collection of sentences and were expanded with data of the assigned CEI databases using a 'monte carlo' method. The words of these corpora define the overall speech recognition vocabulary. A Greek LM for the Athens CEI system is currently in development.
Special emphasis has been given to the pronunciations of words that are not owned by the target language. These words belong exclusively to the data of the CEI databases and appear in event names, location names and food names. The names are often in Italian, Spanish or French and therefore it can not be foreseen entirely, how a foreign speaker would pronounce these words. The current approach tries to approximate different pronunciations of a foreign speaker for the respective language.

3.2 NLU Subsystem

Classer/Parser. For the two-level parsing, the collected application specific corpora for English, Finnish, German and Greek were annotated manually and expanded with data from the assigned CEI databases. The variety of the corpora was increased using a 'monte-carlo' method.
The most profound characteristic of the development for Finnish, German and Greek was how to handle morphological complexity. Since these languages have frequent word conjugation, it was needed to guarantee that relevant forms of a particular lexeme would be properly handled by the system.
The development of the Finnish and Greek classer contained more extensive collection of sentences than the English classer did for instance. As an example, a Finnish location name such as 'klaus kurki' might occur in various forms: 'klaus kurki', 'klaus kurjessa', 'klaus kurkea', 'klaus kurjesta' etc. As Finnish and German, Greek also is morphologically complex. Words like *θεΫτρου* , *θεΫτρων* that are possessive and

accusative forms for singular and plural respectively of the word θέατρο (theatre), were not appearing in the first training corpus. In order to cope with the inflection problem, additional sentences were added to guarantee that all relevant forms are included. In addition, the corpus files containing the units from the database were utilised in the training process: the files did not contain only the non-conjugated forms identical to the database, but various conjugated forms as well. The classer thus learned the conjugated forms similar to the non-conjugated ones. Unlike the various steps of the classer development, the parser required no language-specific treatment.

Canonicalisation. The canonicaliser utilised regular expressions, a word-spotting mechanism and an n-best based pattern matching technique to transform the values of the identified semantic concepts (represented by attribute-value-pairs) to the required format of the backend interface. These techniques also allow a special treatment of morphologically complex partial expressions of an utterance. The transformation guarantees that for instance the Finnish location name 'klaus kurki', 'klaus kurjessa' refer to the same element in the database.

Dialog Modelling. For each CEI implementation a common dialog model across languages was designed for the FDM. The dialog model is implemented using three forms. A MAIN form covers the introductory dialog situation. A set of respective standard messages allows responding to situations such as low speech recognition confidence or silence. The LIST form treats the two semantic concepts of list and query events. It handles the interdependencies of the parameters that are provided by the canonicalised attribute-value pairs for the backend requests. The form defines slots for the event-name, event-type, location-name, date etc. The filled slots determine in accordance to the context whether a backend request can be triggered or an additional information needs to be requested from the user. The third form BYE covers the semantic context of the user intention to finish the dialog and to perform the hang-up procedure.

The model also describes the dialog behaviour for a given utterance in a context dependent dialog flow. It specifically takes into account, whether the current utterance refers to the information provided by previous utterances and the backend application, or the utterance expresses the intention to switch to a new context. In order to cope with the context dependent dialog flow a simple rule-based decision algorithm was implemented that clears the old context according to the following definition:

If the FDM form-level slot values from the current utterance cannot be found within the information provided in the response of the backend to the previous utterance then clear the context (clear all slots), otherwise do nothing.

An example:

User: Give me all music events at karlovasi city hall

System: There is Magic Flute, Karlovasi city hall in Samos. What else can I do for you?

User: When does it take place. ← KEEP CONTEXT

System: There is Magic Flute, Karlovasi city hall in Samos on March 15th. What else can I do for you?

User: And are there any operas next month ← NEW CONTEXT, CLEAR SLOTS

(Here the slots are cleared because of a new context, the location is not kept this time)

System: For opera there are 17 entries. Please choose from the following locations, Athens concert hall, Olympia theatre, Odeon of Herodes Atticus. What else can I do for you?

Language-specific actions also were required for the answer generation. The units in the database were returned in non-conjugated forms; however, in order to make answers natural, conjugated forms would have often been required. Some additional efforts were needed in order to invent appropriate answer templates in which the non-conjugated forms would be in their natural environment.

3.3 Example Dialog

In this section we present an example dialog to illustrate the conversation with the NLU application and the intermediate results of the processing chain. The example was taken when testing the Helsinki City Event Information system. The conversation is started with an initial greeting. Afterwards, the figure draws up a sequence of the user request, the respective class-tree and parse-tree of the two-level parsing, the canonicalised values, if the canonicaliser is invoked, and finally the response of the system by presenting results from the retrieval of the information from the CEI databases. The class- and parse-trees are represented in a flat structure, and show mark up of the identified semantic concepts, whereby the curly brackets enclose a respective semantic concept. As it is shown, in the first utterance the classer and parser identify the request of the user to list all events that suffice the specific event subtype jazz. Since event types need to be canonicalised, the canonicaliser is invoked. The dialog manager evaluates the identified semantic concepts and triggers a back request to retrieve all events that event subtype is jazz. All matching events are presented to the user.
In the sequel the user refers to the second item of the presented event list. The dialog context is furthermore focused on the selected event. More information is requested on the price of the ticket. Finally the user exits the system.

GREETING:	Good afternoon, this is the City Event Information system. How may I help you?
USER:	Tell me about jazz.

CLASS-TREE:	tell_word me_word about_word {E-TYPE jazz_e-subtype E-TYPE}
PARSE-TREE:	{LIST tell null me null about null E-TYPE e-type LIST} . .
CANONICAL:	E-TYPE E_SUBTYPE jazz
RESPONSE:	cloroform from bergen in mother, halla's travels from reykjavik in taidehallin klubi, björn thoroddsen trio from reykjavik in taidehallin klubi and jiri stivin jazzquartet
USER:	could you give information on the second
CLASS-TREE:	could word you word give word information word on word the word {ORD second_ord ORD}
PARSE-TREE:	{QUERY could null you null give null information null on null the null {SELECTOR ORD_ord SELECTOR} QUERY} ._.
CANONICAL:	ORD 2
RESPONSE:	women in jazz from avignon in pravda from march second to march third. What else can I do for you?
USER:	what about the price
CLASS-TREE:	what_word about_word the_word price_word
PARSE-TREE:	{QUERY what null about null the null price q-price QUERY} . .
RESPONSE:	All the events are free of charge. What else can I do for you?
USER:	thanks you bye bye
CLASS-TREE:	thanks word you word bye word bye word
PARSE-TREE:	{BYE thank_polite you_null bye_bye bye_bye BYE} ._.
RESPONSE:	Thank you for calling. Bye.

Fig. 3: Example Dialogue

4 Conclusion and Prospects

In this paper we addressed architecture and components that have been used to built a conversational NLU application. As part of the CATCH-2004 project developments, a telephony-based conversational City Event Information system has been implemented for four languages accessing the city event databases in Athens and Helsinki. This first implementation was done to bootstrap the application based on written data, which does not reflect necessarily the human-machine-dialog like behaviour.

As the refinement methodology is proposing, the bootstrapped application will be used to improve the overall performance. During the user session with the application, the entire human-machine-behaviour is traced. Furthermore, the spoken utterances and the results of the main components including two-level parsing, canonicalisation, respective dialog state and backend responses are recorded. The trace allows an evaluation of the performance of each individual component and also indicates weaknesses in the dialog design with respect to the original user requests. The collected real-life data will be fed into the resources of several components: the collected acoustic data of the spoken utterances will adapt the acoustic models to the domain. The transcribed utterances will enrich the samples to build the respective language model, and the resources for the two-level parsing.

The development prospects of the project are three-fold. Firstly the refinement of the developed resources for the CEI application. Secondly the transition of multi-language resources to multi-lingual resources for the respective components, whereby the system handles all four languages with the same components at the same time. And thirdly the development of two new applications of wider scope: an Olympic Sports Event application and a Programme Guide Information System.

5 References

[1] K. A. Papineni, S. Roukos, and R. T. Ward, *Free-Flow Dialog Management Using Forms*. Eurospeech 99, Budapest, Hungary, 1999.

[2] 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.

[3] Berger, A., Della Pietra, S., Della Pietra, V., *A Maximum Entropy Approach to Natural Language Processing*, Computational Linguistics, Vol. 22, No. 1, pp. 39-71, March 1996.

[4] 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.

[5] CATCH2004 European Research Project – Official Website: http://www.catch2004.org