

Implementing a Natural Language Conversational Interface for Indian Language Computing

Rahul Jindal

Infosys Technologies, Mohali

rahul_j@infosys.com

Rohit Kumar

Ritvik Sahajpal

Sanjeev Sofat

Shailendra Singh

Department of Computer Science and Engineering, Punjab Engineering College

Sector 12, Chandigarh, INDIA 160012

rohit@iiit.net, rsahajpal@quark.co.in, chestasofat@yahoo.com, shailendra_sing@yahoo.com

Abstract ? Indian Language Computing is lagging behind due to several reasons, primary amongst those being unavailability of suitable interfaces for human computer interaction. Natural Language Conversational Interfaces (NLCIs) can play the all-important role in improving the usability of interfaces for the Indian masses. In this paper a NLCI incorporating speech and dialog is presented.

I. INTRODUCTION

Natural Language Conversational Interfaces (NLCIs) use the natural modes of interaction among humans like speech and dialog to provide better interfaces for human computer interaction. These NLCIs are motivated by the need to bring human computer interaction as close to inter-human interaction as possible.

There are two basic challenges involved in development of NLCI. First – a system needs to be able to process and respond to the natural language conversation inputs from the user. The response should be by an output in natural language and/or by a system level action. Second – the modes of interaction with the user should be similar to inter-human interfaces such as spoken dialog and facial expressions. There are several issues involved in each of these basic challenges.

Through this paper, we describe the development of a NLCI for Indian language computing. The challenges involved in the development of NLCIs [1] are met through convergence of existing technologies in the domain of Dialog Management and Spoken Language Generation. A Text In Speech Out interfaces is implemented.

II. THE COMPUTING TIME COMPANION MODEL

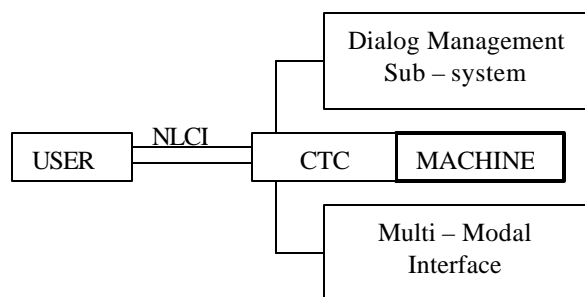


Fig. 1 Computing Time Companion Model

In fig. 1 the basic model of a Computing Time Companion (CTC) is shown. The CTC stands between the user and machine and provides a natural language conversational

interface for human computer interaction. The CTC makes the interaction with the machine closer to inter-human interaction. For the user the CTC acts as a human like companion helping him in accessing the machine's services – hence the name Computing Time Companion.

The CTC's ability to provide human like interaction is provided through two basic subsystems: the Dialog Management subsystem and the Multi Modal Interface.

III. DEEPTI: IMPLEMENTATION OF COMPUTING TIME COMPANION

To provide an instantiation to the CTC model proposed in Section II, we have implemented a NLCI named Deepti.

A. Characteristics of Deepti

- 1) *Human like Conversational ability:* Deepti is capable of intelligently interacting with the user through natural dialog. The natural dialog is user controlled instead of being a fixed set of prompts to which the user replies in a specified format.
- 2) *Text In Speech Out Interface:* Deepti's natural dialog response to user stimulus has been made multi modal by providing the responses both as readable text and as speech. The inputs from the user remain in text format.
- 3) *Controls the Machine for the User:* The interface controls the machine i.e. the operating system for the user by generating system level actions like opening up a notepad in response to user's explicit requests like "I want to type down some notes" or implicit mentions like "I have got an idea".
- 4) *Support for Hindi:* We have concentrated our efforts on implementing the Deepti NLCI for one Indian language – Hindi. It should be mentioned that the approach followed is quite general for implementation of a similar NLCI for other Indian languages.
- 5) *Easy Modifiability for new domains:* The architecture of Deepti as discussed ahead and the technologies we have used are easily modifiable to incorporate knowledge about new domains and to adapt to new applications of Deepti.

B. Architecture of Deepti System

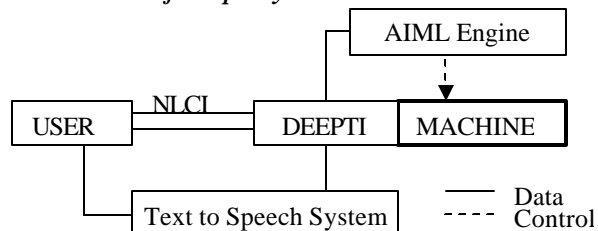


Fig. 2 Architecture of Deepti System

The block diagram of architecture of the Deepti system is shown in fig 2. The derivation of this architecture from the CTC model shown in fig 1 is clearly observable. The dialog management subsystem is replaced by the Artificial Intelligence Markup Language (AIML) engine. The multi modal interfaces currently comprise of an Indian Language Text to Speech System only.

The user and Deepti-the interface to the machine interact through natural dialog strictly following a protocol of taking turns in communicating alternately.

C. Standard and Encoding for Hindi: Use of Romanized Hindi

To avoid effort involved in learning the layout of a hindi keyboard, we have used Romanized Hindi for the textual inputs and responses. Romanized Hindi in itself has no standard for encoding each grapheme in Hindi. Due to this there is a lot of flexibility for the user to type in the words in a natural way. For example the word हमारा (meaning: our) can be written in different ways like hamara, hamaaraa, hamaara, hamaraa in Romanized Hindi.

IV. ARTIFICIAL INTELLIGENCE MARKUP LANGUAGE

Due to successful use of Artificial Intelligence Markup Language (AIML) [2] in several experiments and applications, we have chosen AIML engine for use as the dialog management subsystem for the Deepti system. Fig. 3 shows the basic architecture of AIML Engine. [3][4] discuss the architecture and pattern matching algorithms used in AIML Engine in detail.

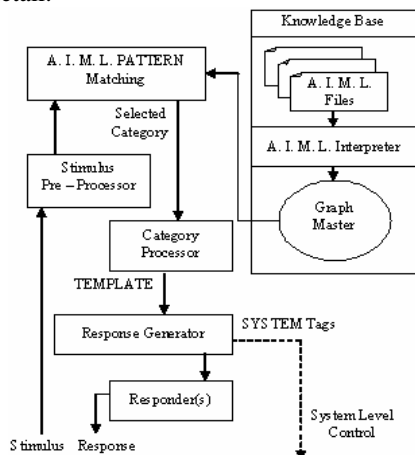


Fig. 3 Architecture of AIML Engine

B. Approaches to Build AIML Knowledge Base

The first approach to building AIML Knowledge Base is Anticipatory. The botmaster enumerates all the most common inputs and writes a suitable response to these inputs to form the categories. This approach can be made more convenient if two botmasters engage in a dialog with one of them playing the role of a user and the other the role of the machine and processing the dialog to generate categories.

The second approach is based on backward looking log file analysis. The botmaster adds knowledge to provide correct response to the inputs which were incorrectly answered, by analysing the interaction logs of the bot.

V. INDIAN LANGUAGE TEXT TO SPEECH SYSTEM

To provide a multi modal response to the user comprising of both written and spoken form of dialog, we have built a Text to Speech (TTS) system for Hindi based on Festvox Framework. A diphone based voice has been built.

A. Building a diphone database for Hindi

The major task involved in building the Hindi TTS using Festvox framework is that of building diphone database [5].

- 1) *Definition of Phonetset*: A Hindi phonetset comprising of a silence, 12 vowels and 32 consonants has been defined.
- 2) *Listing the Diphones*: Based on the total 45 phones in the definition of Hindi Phonetset a list of all possible diphones was generated and later some of the phonotactically invalid diphones were manually deleted from the list.
- 3) *Recording of Carrier Words*: The carrier words are recorded at a rate of 16000 samples per second.
- 4) *Labeling the Recording*: The recordings were manually segmented at phone boundaries.
- 5) *Extracting Features*: Pitchmarks and pitch synchronous linear prediction coefficients are extracted and stored offline.

B. Text Processing Modules for Hindi

A pronunciation lexicon comprising of all the words in our AIML templates along with their pronunciation has been built. To generate pronunciation for unseen words, a set of carefully designed letter to sound rules has been written.

VI. POSSIBLE APPLICATIONS OF DEEPTI SYSTEM

Primarily the Deepti System could serve as the interface for Information and Service Kiosks being developed for rural and suburban India.

The Deepti System could be deployed as an information dispensing system at shopping centers, commercial banks, post offices, railway stations, airports, tourist spots, etc. Our NLCI also serves as the primary interface to computer for a layman who requires using only certain basic application like emailing, taking notes, writing letters, etc.

VII. REFERENCES

- [1] V. Zue, "Conversational Interfaces: Advances and Challenges," *EUROSPEECH*, 1997, pp 9-18
- [2] Artificial Intelligence Markup Language, <http://www.alicebot.org/alice/aiml.html>
- [3] R. S. Wallace, "The Anatomy of ALICE," *Intl.LISP conference*, 2002, <http://www.alicebot.org/anatomy.html>
- [4] R. S. Wallace, "AIML Pattern Matching Simplified," <http://www.alicebot.org/documentation/matching.html>
- [5] A.W. Black and KA. Lenzo, "Building Voices in the Festival Speech Synthesis System", Edition 1.2:beta, 2000 <http://cstr.ed.ac.uk/projects/festival/docs/festvox/>