Vaidya: A Spoken Dialog System for Health Domain

Prathyusha Danda* **Brij Mohan Lal Srivastava*** **Manish Shrivastava**
International Institute Information and Technology, Hyderabad
{danda.prathyusha, brijmohanlal.s}@research.iiit.ac.in
m.shrivastava@iiit.ac.in

Abstract

In this paper, we introduce Vaidya, a spoken dialog system which is developed as part of the ITRA[1] project. The system is capable of providing an approximate diagnosis by accepting symptoms as free-form speech in real-time on both laptop and hand-held devices. The system focuses on challenges in speech recognition specific to Indian languages and capturing the intent of the user. Another challenge is to create models which are memory and CPU efficient for hand-held devices. We describe our progress, experiences and approaches in building the system that can handle English as the input speech. The system is evaluated using subjective statistical measure (Fleiss' kappa) to assess the usability of the system.

1 Introduction

Healthcare is a basic need of any sustainable society and has advanced many folds since the advent of technology. But effective medical diagnosis still remains inaccessible to the rural population in developing nations. The main reasons for this are scarcity of skilled healthcare staff in countries like India[2] and minimal Internet connectivity. Dialectal variations in rural areas render remote diagnosis methods like telemedicine, ineffective. Using spoken language technology, we can fill this gap and bring the state-of-the-art healthcare at the hand's reach of almost everyone. This paper presents a spoken dialog system which targets problems like understanding low-resource languages, inferring diagnosis from medical ontologies, capturing intent of the user and working with resources having limited memory and computational power such as a handheld devices.

Spoken dialog systems (SDS) have been an active area of research for the past few decades. But a large body of work has gone in developing SDS for English. There are several active systems currently in use for travel and healthcare in English. Research projects in India focus on understanding the linguistic structure of Indian languages and to make them easily representable in digital form. Structural analysis of languages coupled with SDS can create viable solutions for healthcare. There is a huge necessity of SDS in Indian healthcare systems since 1) medical knowledge is readily available through well-crafted disease ontologies which can be easily queried and 2) the mortality rate in rural areas is much higher due to lack of advanced diagnosis[3].

Most of the recent language technologies being developed currently are feasible on a standard computer with a good internet connectivity. Lately hand-held devices are gaining a lot of power both in case of memory as well as computation and are available at a reasonable cost. This has made these devices very handy to a large extent of population. Keeping the pervasive nature of mobile phones in mind, we can design solutions that are available to a large section of society. We can even penetrate the rural areas since the core technologies of dialog systems do not expect high literacy among its users and even work for languages that have no written form. Such interfaces where speech is used as underlying modality hold great promise as a preferred choice.

Main contributions of this paper are as follows:

- Domain-independent dialog flow structure built to scale to multiple languages and do-

*equal contribution

[1] Information Technology Research Academy http://itra.medialabasia.in/

[2] https://data.gov.in/catalog/rural-health-statistics-india-2014

[3] http://www.who.int/countries/ind/en/

mains

- Low resource adaptation of Large Vocabulary Continuous Speech Recognition (LVCSR) to improve Indian language speech recognition

- Creation of healthcare domain ontology compatible with domain-independent dialog structure

This paper is organized as follows: The following section gives an overview of related literature. In section 3 we furnish the details of data that is collected as well as the open source data which is used to build different modules. In section 4 we describe the overall architecture of Vaidya with subsections describing the modules of the spoken dialog system. Section 5 that follows, concludes the paper and planned future directions are mentioned.

2 Related Works

Since early 1970s there have been several clinical decision support systems (CDSS), like Internist-I (Miller et al., 1982), Mycin (Shortliffe, 2012), etc. which help in clinical knowledge management and assist the healthcare providers to diagnose with desirable accuracy. But these systems are not directly accessible and operable by the patients. Bickmore (Bickmore and Giorgino, 2006) and Barahona (Rojas-Barahona, 2009) wrote a detailed account for creation of such systems and the methodologies involved that can be used for construction as well as evaluation. Sherwani (Sherwani et al., 2007) worked on the ideas similar to the current work where community health workers and dialog system technologies were collaboratively used in order to provide healthcare to the low-literate sections of Pakistan.

3 Data description

Medical knowledge is vital for the development of system hence we used the human disease ontology[4] created by Institute of Genome Sciences. This ontology is transformed into a domain-specific knowledge-base which coordinates with dialog manager to determine the state of the dialog and further actions. The ontology contains diseases as classes and relationships such as "has_symptom" which help create a knowledge-base of diseases and their corresponding symptoms. All those diseases which have minimum of three symptoms were taken into account with no constraint on upper limit. All the diseases which have two symptoms were segregated and were checked manually to see if any popular diseases are present and then were added to the ontology. The plural forms of the symptoms were replaced with the singular form to maintain consistency throughout the knowledge-base. E.g. lymph nodes $\rightarrow$ lymph node. Verb forms were replaced with the respective stems. E.g. coughing $\rightarrow$ cough. After these modifications the ontology consists a total of 560 diseases and 623 symptoms. This knowledge can also be used to create language models and in collection of relevant speech data.

In order to train the acoustic models with speech specific to Indian languages, we use single-speaker CMU Arctic database which is recorded in Indian English. We also collected speech samples for English, Telugu and Hindi (approx. 100 sentences each) from students of IIIT-H.

4 System Architecture

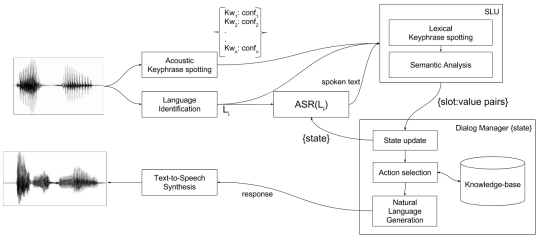

Figure 1: Vaidya architecture

The general framework of Vaidya is illustrated in Figure 1. It consists of modules like ASR, Dialog Manager, Natural Language Understanding, Knowledge-base and Text-to-Speech Synthesis. ASR is naturally the starting point of the dialog flow, where input is collected in form of speech signal and converted to text. This text is accepted by the dialog manager, which performs natural language understanding over it to understand the semantics of the spoken input. It queries the knowledge-base to check the state of the dialog flow and retrieves relevant information. Finally, a suitable response is generated by the text-to-speech synthesis module to let the user know the

[4]http://disease-ontology.org/about/

state of the dialog and prompts to expect required responses. During the development of the system, we encounter several challenges. One of them is speech recognition in Indian languages which can be implemented as keyword spotting using articulatory gestures as cardinal units (B. M. L. Srivastava et al., 2016). Currently the system focuses on building recognition engine for English with Indian accent by adapting the existing acoustic models towards Indian language speech. After recognition, the output is normalized and represented as a text which can be parsed by keyphrase spotter in order to interpret the embedded legal symptoms as per the ontology. These symptoms are queried against the knowledge-base and a list of probable diseases is generated. Additionally, the system also takes pictures from inbuilt camera on mobile as input and the future extension is to normalize them to legal symptoms. This input will help in classification of wounds or rashes.

4.1 Multilingual Automatic Speech Recognition (ASR)

Usually the knowledge is represented in form of concepts, specifically, structured text such as Resource Description Format (RDF) or as semantic networks of relations such as ontologies. Therefore in order to query this knowledge it is of paramount importance to convert spoken speech signal to lexical units. The final aim of this module is to convert speech signal spoken in any language to corresponding text. But each language follows its own set of sound units which vary largely based on geography. Hence it is not feasible to create a speech recognizer which understands all the languages perfectly. There have been several attempts to create a multilingual speech recognizer based on two approaches. A two-pass method where a language identification system (B. M. L. Srivastava et al., 2015) is followed by several ASR modules as shown in figure 2. This module segments the speech signal based on different languages present in the signal. Then each segment is fed to its corresponding ASR for transcription. Another approach (Figure 3) is to train the ASR over a multilingual speech corpus which covers most of the phones that are expected to be encountered in the test signal.

The performance of ASR can be improved by tuning the acoustic models, pronunciation dictionary and language models. In order to train acoustic models we used the phonetically tied model provided by Pocketsphinx (Huggins-Daines et al., 2006) as the base model and adapted it using relevant domain-specific data. We used CMU Arctic database and the collected medical speech samples to adapt the model for Indian languages. The word error rate (WER) obtained when we adapted the model by different data is mentioned in Table 1.

Pronunciation dictionary is manually created for all the medical words that are expected to be encountered as speech input. We used US English phones to approximate the pronunciation of each word. Language modeling of the word sequences is important based on the context that is being discussed currently as part of that dialog. Therefore, three types of language models were constructed. 1) Generic trigram language model for free-form open ended conversation; 2) symptoms language model in JSpeech Grammar Format (JSGF); and 3) binary language model for affirmative / negative responses.

4.2 Dialog Manager

This module is the central part of the system which maintains consistency between the utterances. For each conversation, it creates a context object which stores several parameters needed to proceed with the dialog. These parameters are state-dependent. Current system works with 7 states namely,

- greet state
- ask_symptoms state
- diagnosis state
- disease_details state
- symptoms_details state
- disease_enquiry state
- first_aid state

Each of these states has a global set of flags which help the dialog manager to track the progress of dialog. They can also change the language model based on the requirement. E.g. a state which is expecting a boolean affirmative / negative response need not search in a huge generic language model. System successfully works on laptop as well as a low-end Android mobile and performs in real-time.

Initially, the greet state takes user input and determines the goal of the conversation. It finds out the domain which has to be selected and queried. This is for the purpose of scalability since there can be several other domains which

Acoustic model	Word Error Rate (WER %)
default model (en-us-ptm)	39.77
default + adapted using CMU Arctic	39.79
default + adapted using IIIT-H speech samples	**24.54**
default + adapted using CMU Arctic + IIIT-H speech samples	24.56

Table 1: WER for each acoustic model

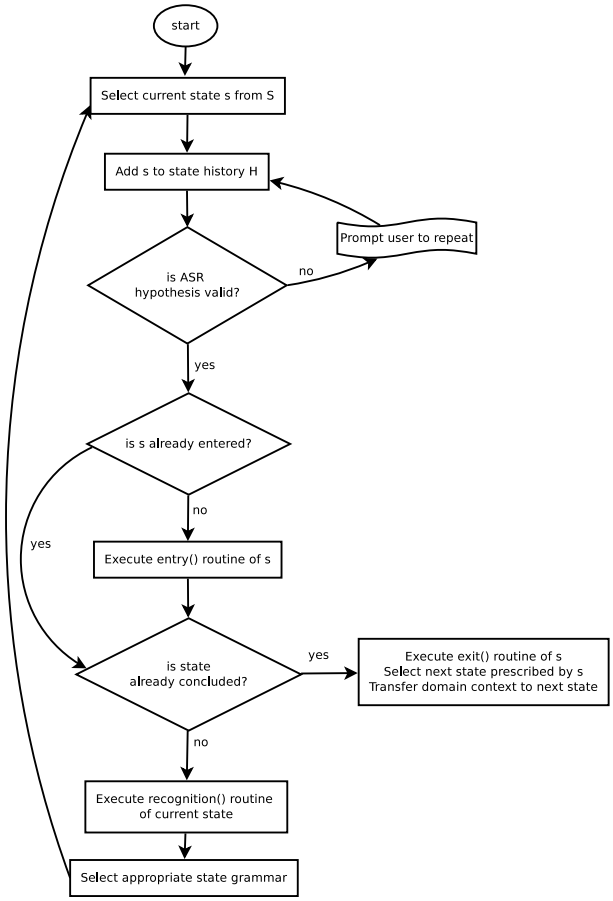

Figure 2: Domain-independent dialog flow

can be integrated within the framework. It also determines out- of-domain utterances and rejects them. ask_symptoms state is selected when user wants to describe their symptoms for diagnosis. This state accepts natural language speech and looks for symptoms that can be present in the transcribed text. Once the user has exhausted their list of symptoms, the system enters diagnosis state. This state queries the knowledge base and narrows down to a list of most probable diseases. This state further finds out the symptoms corresponding to these diseases and assigns a score to each one of them which measures the number of times each symptom occurs in the disease list. We follow the algorithm mentioned as Algorithm 1 in order to reduce the list down to one or zero disease. Depending on the user's response the system may prompt another query or give the diagnosis. The user has an option to ask what a particular symptom means, to which the the system will give information about the symptom by going to symptom_details state and then automatically goes back to diagnosis state to proceed with the diagnosis. We narrow down to one or zero disease by querying the user with that symptom, which we get from following Algorithm 1. The user is presented with the result of diagnosis. Finally the dialog enters disease_details state if the user wants to know more about the disease, else he is prompted with the greet state.

User can continue with other functionalities of the application which include first aid, disease inquiry and diagnosis. The first aid flow of the application requires the user to mention the type of help required. The system will then prompt the steps to be followed and guides the user through the interactive process of treatment where user can provide speech and images as input.

User can also inquire about a disease just by providing the disease name. The system will give the definition, symptoms and causes of the disease in form of text as well as speech for advanced accessibility. In another scenario, user may want to know if they have certain disease and the system will interactively find the symptoms and causes particular to that disease.

5 Evaluation & Results

We incorporate Fleiss' kappa statistical measure in order to assess the usability of submodules and the entire system. Evaluation is conducted by employing 10 subjects with varying age groups and gender. All the subjects are Indian non-native English speakers. Each subject was assigned 5 randomly selected diseases which are unique to them. Ratings were assigned by the subjects based on following five parameters.

Algorithm 1 Algorithm for diagnosis

procedure DIAGNOSIS
  $D \leftarrow$ (d_1, d_2, ... , d_k)
  $S \leftarrow$ ($s_1 : n_1$, $s_2 : n_2$, ... , $s_m : n_m$)
  while $len(D) > 1$ **do**

$$\tilde{s_i} = \underset{\tilde{s_i}}{\operatorname{argmin}}(len(D)/2 - n_i) \quad (1)$$

    prompt user to ask if he/she is observing $\tilde{s_i}$
    mark $\tilde{s_i}$ based on user response
    remove corresponding diseases from D
  return D

1. Ease of use (1 - System is confusing and misleading, 5 - System is seamless and well-guided)

2. ASR performance (1 - Recognizes nothing correctly, 5 - Recognizes everything correctly). ASR performance is also quantitatively measured as fraction of correctly recognized utterances for objective analysis.

3. How often does the system digress from the main goal? This measure analyzes if the user is being interrupted by annoying information in the middle of dialog.

4. Success rate. This measures that how many times the system reaches correct goal.

5. Steps taken to reach the goal. The system is designed to optimize the total number of steps in order to complete the dialog. In ideal case, system must halve the disease search space with each question it asks, so it must converge to a disease in no more than $\log_2 N$ steps, N being total number of symptoms. Since the dialog flow does not follow the ideal scenario due to speech recognition errors and uneven distribution of diseases in symptom space, system may take more than $\log_2 N$ steps to converge. We objectively measure this and incorporate it into the evaluation criteria.

Table 5 shows the subjective evaluation results by 10 users pertaining to 5 measures. The rating is from 1 to 5 for each measure, 1 being the lowest rating and 5 being the highest. 86% of the dialogs led to successful completion with correct diagnosis. The last column of Table 5 refers to the average number of steps or questions asked by system in order to complete the diagnosis. Minimum number of steps to complete the diagnosis is 0 and maximum number of steps is 16. On average, system requires 3.8 number of steps to reach the diagnosis.

We compute the overall rating of the system using Fleiss' kappa measure (κ).

Overall agreement	0.50
Fixed marginal kappa	0.15
Free marginal kappa	0.37

Table 2: Kappa measures

6 Conclusion & Future Work

We described the general framework of Vaidya, which accepts speech as input from the user and guides her to diagnose symptoms and possible diseases. We further describe its various components and subsystems and the resources we used to build each of one them. The overall system successfully works on laptop as well as a low-end Android mobile and performs in real-time. This system will act as the baseline for further versions of Vaidya. There is scope for improvement at symptom recognition level and capturing the semantic invariance of user's intent in the framework. Extensive speech resources are being collected in form of medical conversations which will enable the system to improve and extend the recognition capability. These resources can also be used to model the dialogs and learn to adapt based on the dialog act history and the state which is currently active. We also need to formulate a general evaluation strategy which can provide objective as well as subjective measure of overall system performance.

Acknowledgments

Authors would like to thank ITRA for supporting the project financially.

References

T. Bickmore and T. Giorgino. 2006. Health dialog systems for patients and consumers. *TJournal of biomedical informatics*, 39(5):556–571.

D. Huggins-Daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar, A. Rudnicky. 2006. Pocketsphinx:

User index	Ease of use	ASR performance	Consistency	Success rate	# Steps
1	4	5	4	5	4
2	4	4	4	5	4
3	4	3	4	5	4
4	4	5	5	5	5
5	3	4	4	5	5
6	5	4	5	5	5
7	4	4	5	5	5
8	5	4	5	5	4
9	5	4	3	5	4
10	3	2	4	5	4

Table 3: Subjective evaluation results over 50 diseases

A free, real-time continuous speech recognition system for hand-held devices. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 1, pages I–I.

R. A. Miller, H. E. Pople Jr, and J. D. Myers. 1982. Internist-i, an experimental computer-based diagnostic consultant for general internal medicine. *New England Journal of Medicine*, 307(8):468–476.

J. D. Osborne, S. Lin, W. Kibbe, L. Zhu, M. Danila, and C. Rex. 2007. Generif is a more comprehensive, current and computationally tractable source of gene-disease relationships than omim. *Bioinformatics Core, Northwestern University Technical Report*.

L. Rojas-Barahona. 2009. *Health care dialogue systems: practical and theoretical approaches to dialogue management.*PhD thesis, University of avia, Pavia, Italy

J. Sherwani, N. Ali, S. Mirza, A. Fatma, Y. Memon, . Karim, R. Tongia, and R. Rosenfeld. 2007. Healthline: Speech-based access to health information by low-literate users. *Information and Communication Technologies and Development, 2007. ICTD 2007. International Conference on*, pages 1–9. IEEE, 2007.

E. Shortliffe. 2012. Generif is a more comprehensive, current and computationally tractable source of gene-disease relationships than omim. *Computer-based medical consultations: MYCIN*.Elsevier.

B. M. L. Srivastava, H. K. Vydana, A. K. Vuppala, and M. Shrivastava. 2015. A language model based approach towards large scale and lightweight language identification systems.

B. M. L. Srivastava and M. Shrivastava. 2016. Articulatory gesture rich representation learning of phonological units in low resource settings. 4th SLSP 2016.