Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 8, Issue, 06, pp.32456-32460, June, 2016

INTERNATIONAL JOURNAL
OF CURRENT RESEARCH

ISSN: 0975-833X

RESEARCH ARTICLE

IMPLEMENTING VOICE INTERACTIVE HOME AUTOMATION

Saim Ahmad, *Kaushal Mundada, KunalPhaltane and NiharRanjan

Department of Computer Engineering Sinhgad Institute of Tech and Science, SavitribaiPhule Pune University

ARTICLE INFO

Article History:
Received 23rd March, 2016
Received in revised form
29th April, 2016
Accepted 26th May, 2016
Published online 15th June, 2016

Key words:

NLTK, NLP, Parsing,
Home Automation.

ABSTRACT

The science of home automation took roots in the late eighteen hundreds. However, the practical implementation were realised not until early twentieth century. Voice interaction to communicate with computer applications is a relatively old idea initiated by Bells labs in as early as 1930's. The use of voice in controlling elements in the home has not been extensively explored. There exist home automation systems that respond to voice, but these systems have a limitation. These home automation systems identify only a fixed close set of pre-specified voice commands. Such systems do not generalise well for speech commands that are not included in the pre-specified set of identified voice commands. The project undertaken by us attempts to push the frontier of home automation by including natural voice interactivity within the home automation system. In other words, the aim is to understand a command even if it is presented in one of its paraphrases. The report outlines the methodology we plan to undertake and the progress we have made so far.

Citation: Saim Ahmad, Kaushal Mundada, KunalPhaltane and NiharRanjan, 2016. "Implementing voice interactive home automation", *International Journal of Current Research*, 8, (06), 32456-32460.

INTRODUCTION

For describing previous work, we will present the earlier progress dichotomised into control of devices in home and voice processing. The earliest record of remote controlling devices goes back to 1898; a patent issued to Nikola Tesla for remote controlling of vessels and vehicles. Gradually, with the proliferation of micro controllers in the late nineteenth century, the reach of the so-called smart devices was widened. This was a major reason for the emergence of Internet-of-Things which in turn was a major boost to the commercialisation of home automation systems. In regards to the speech recognition, much of the pioneering progress happened through Bell labs and DARPA funding. A variety of different statistical models and methods like hidden Markov models, n-gram, log linear models among others tackle problems in speech processing (Jurafsky and Martin, 2000). For a systematic review of the stages in natural language treatment, refer (Michael Collins, 1999). The project is proposed to support two types of voice inputs-commands and queries that control the devices in the home. Commands take the form of an imperative statement where the user specifies an action. For instance, commands could be any one of these, "Turn off the lights in the hall" or "Switch off the lights when my father leaves". Queries, on the other hand, are used to enquire about aspects related to home. A few examples of queries would be, "Where in the house can I find my mom?", "Where is dad?", "Where is my friend?", "How many people are there in the guest room?", "How many rooms are unoccupied?" Note that the queries and command do not conform to a strict format. These are all supported commands and the system can handle these input cases and respond to them as required.

System architecture

This section will proceed to explain the system overview and mention the functionalities of the components in some detail. The high level architectural view can be illustrated using the Figure 2. A mobile application is the first point of contact for the user to interact with the system. The user's mobile phone is used for two reasons- firstly, to accept speech sample from the user and secondly, to track the location of the user carrying the mobile. We will proceed to the discussion of the system by looking at data flow within different components of the architecture. We begin with the stage of user interacting with the system using mobile phone and end our discussion with the system performing the desired action. A simple walkthrough will be sufficient for understanding the high level working. The user speaks a wake up phrase to activate the voice receptive system in the phone. Once activated, the user speaks the command or query into the microphone. This voice sample

***Corresponding author: Kaushal Mundada,**
Department of Computer Engineering Sinhgad Institute of Tech and Science, SavitribaiPhule Pune University

is converted to text using third party API's. The converted text is then passed on to the central language processing module for further processing. The language processing unit performs three important tasks namely tokenization, POS tagging and parsing. Once parsing is complete, we apply hand written rules to understand the informal query or command. After that task, we look up the knowledge base that contains information about the home users. The knowledge base stores the home owner's current locations which are updated on real-time basis. As for the mathematics and structure of the informal commands and their corresponding machine understandable commands, we have created a format that is completely structured. This structured representation of the command makes it machine understandable. The structured representation is a typical JSON object with nested elements that specify what action is to be performed. In a typical working scenario, the incoming text is parsed first. From the parse tree generated, the main action word is extracted by traversing the parse tree and examining a node for the VB (Verb base) tag.

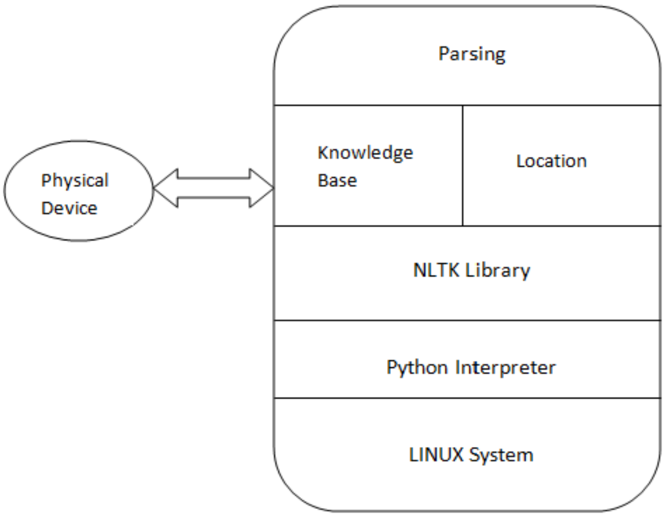


Fig. 1. Server Module

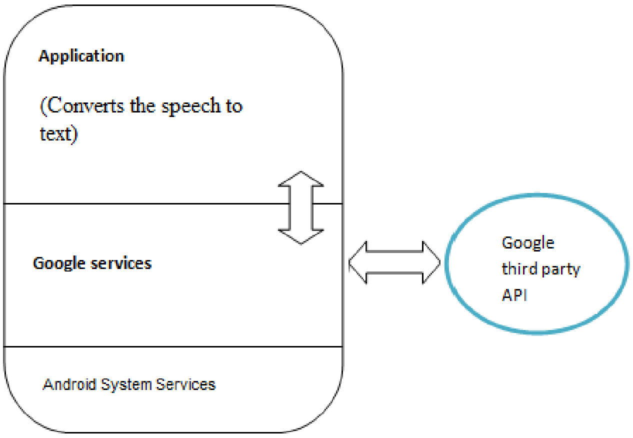


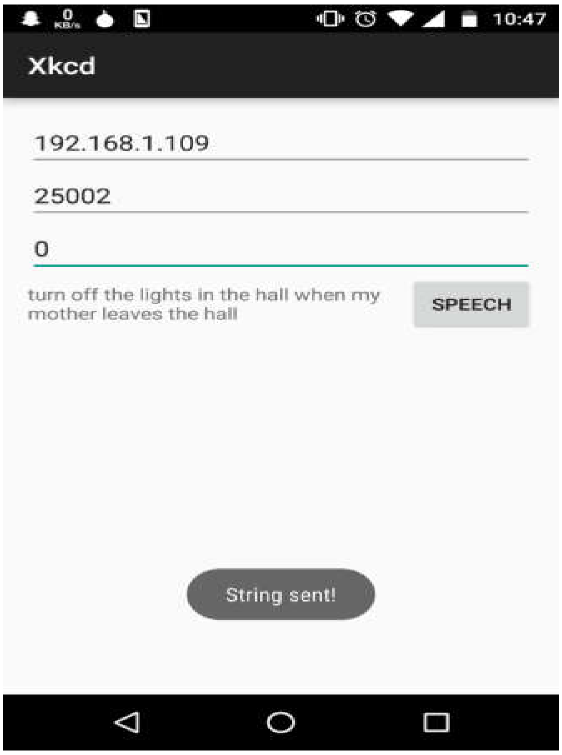
Fig 2. Client Module

Once the verb base form is found, we take that leaf node of that tag giving us a word that is the main verb in the sentence. The word is passed to the module that emits all the semantically similar words of that verb. Using all the emitted responses from the word net, the module checks if the word corresponds to a predefined action verb. If it does, then the parse tree is referred again to extract the object on which the main verb acts. Using a similar process and a set of hand written rules, the object is extracted from the parse tree. The module also examines if the time to perform the action is specified in the informal command. If it is specified, then the one of the elements in the JSON object is the time specified to perform that action.

Proposed Methodology

The mobile application which is installed at user end i.e. in all house members' mobile phone and that will be the first point of interaction with the system. The user will fire his query/command verbally by pressing the speech button. The mobile application is using Google API to convert the speech to text and once the speech is received in text format, it is sent to server side over the network for further processing. While sending the query, the user also needs to send his ID to server and once the string is sent, a message is displayed that the string is sent. At the server side, it continuously keeps listening for the string and once it receives one, it extracts all the important details out of it. It first looks for the user who fired that query and that is easily done by user ID. Later the system forward the textual string to the disambiguation module, the disambiguation module classifies the type of the textual string. The type of the string can either be a command or a query. A command is an imperative statement that instructs the house to perform a certain action whereas a query is a question that asks for particular information or a statistic about the house. An example of a command would be a sentence like, "Turn off the lights in the evening". On the other hand, an example of a query would be, "How many people are there in the house". When we talk about query, it can be classified it either as room usage query or location query. And all the other things can be considered as a command. Query always starts with the H-question or a WH-question word like "How", "Where". Example for this can be like 'How many rooms are empty' or 'Where can I find my father'. The location module takes in the sentence and builds a parse tree of that sentence. Upon building the parse tree, the subject of the sentence is determined that is, the person whose location is to be found is determined. It must be noted that the subject of the query is not always in the normalised form. It can be a direct name like "Kunal" in the case of "Where is Kunal" however; it can also be an indirect reference in the case of "Where is my son". In the latter case, the relationships between the speaker and her family members must be elicited from the knowledge base maintained in the system. Once the normalised subject of the sentence is extracted, the knowledge base is looked up for the real time location of the person and the response is returned. Now, talking about the commands that can be fired includes many things like "Turn off the lights when I leave my room" or "Turn off the fan at 6 p.m." The processing these commands are done again by parsing the sentence and making the parse tree and analysing certain parameters like "what action need to be performed" i.e. turn on/off . Second thing which need to assess is "on what action need to be performed" i.e. the device-fan/light. The third tuple will be the time if specified by the user and if not specified then -1 is taken in tuple. Fourth and fifth tuple will be room id and then user id i.e. in which room these action need to be performed and user id will be the person firing these commands. And the last tuple will be the condition tuple i.e. "perform action when I leave the room". So at last the respond is generated by the system with all the tuples and accordingly the action is performed.

Once the entire tuple is generated, it is analysed and then sent to the microcontroller. Once the tuples are received by micro-controller it will analyse the tuple and will take them as input for arduino program. There will be simple program which will control all the hardware and the required action will be performed as per the tuple generation. Once the action is performed a message will be sent back to processing unit and from there back to the user's mobile displaying the message that the required operation has been performed.



Xkcd

192.168.1.109

25002

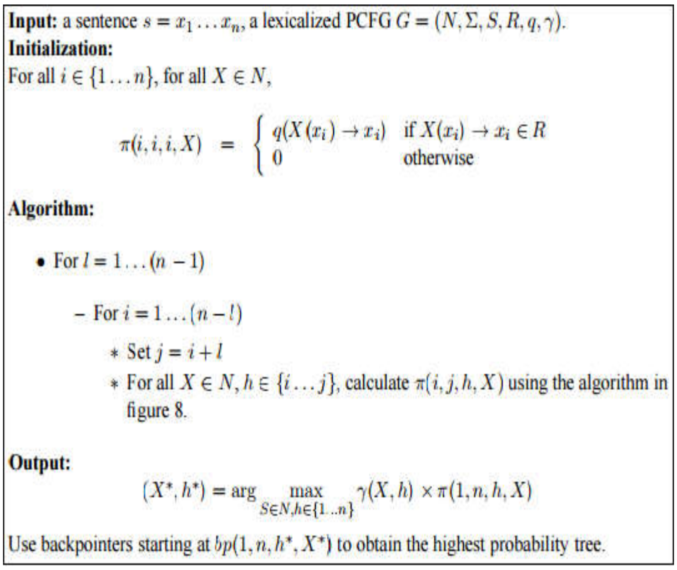
0

turn off the lights in the hall when my mother leaves the hall

String sent!

Implementation

The programming language used for the project is python as majority of the work is performed using NLTK library for which support is available in python. The knowledge base is built in a MySQL relational database. In the knowledge base, information about home owners like the location of each members of the home, the room assigned to them and their familial relations are stored. The location tracking is performed using Bluetooth triangulation. For detailed information on how to perform refer (AdwaitRatnaparkhi, 1997). This location information is updated on a real time basis with a python script that runs after every 20 seconds. For POS tagging we are using the Viterbi method (NidhiAdhvaryu and PremBalani, 2015). This is the tagger is available in the nltk tagger package. For parsing, we are using a PCFG based parser that uses CKY algorithm. The rules of the PCFG are learned from the Penn tree bank. The algorithm for the CYK parsing is shown below. (Michael Collins, 1999)



```
Input: a sentence $s = x_1 \ldots x_n$, a lexicalized PCFG $G = (N, \Sigma, S, R, q, \gamma)$.
Initialization:
For all $i \in \{1 \ldots n\}$, for all $X \in N$,
    $\pi(i,i,i,X) = \begin{cases} q(X(x_i) \rightarrow x_i) & \text{if } X(x_i) \rightarrow x_i \in R \\ 0 & \text{otherwise} \end{cases}$
Algorithm:
  • For $l = 1 \ldots (n-1)$
      – For $i = 1 \ldots (n-l)$
          ∗ Set $j = i + l$
          ∗ For all $X \in N, h \in \{i \ldots j\}$, calculate $\pi(i,j,h,X)$ using the algorithm in figure 8.
Output:
    $(X^*, h^*) = \arg \max_{S \in N, h \in \{1 \ldots n\}} \gamma(X,h) \times \pi(1,n,h,X)$
Use backpointers starting at $bp(1,n,h^*,X^*)$ to obtain the highest probability tree.
```

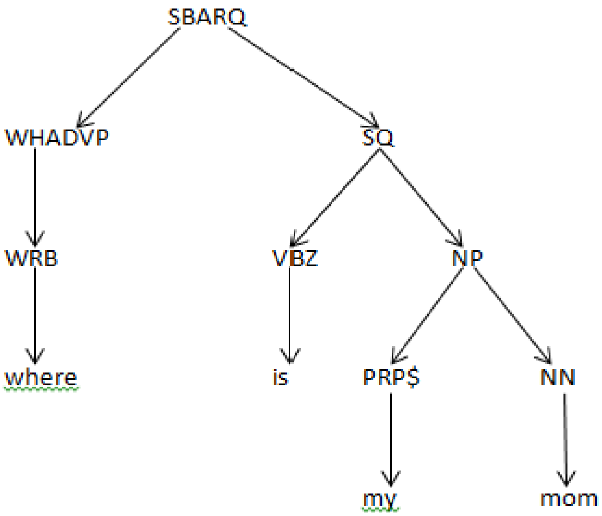


Fig.3. Parse Tree

So far, implementing half of the project is complete. Concretely, the first point of interaction of user with this system that is mobile application with both manual and voice interaction facility, partial knowledge base and the query processing module is complete. What remains to be implemented is the command processing unit which takes in an informal command and performs the corresponding action. The query processing sub-module in the language processing module is completely implemented. It can answer questions like, "How many people are there in the hall", "Where can I find my father", "Where is my mother", "Do you know where is my daughter" and so on. For training the parser, the corpus that we used was the question tree bank (Cutting *et al.*, 1992) and the parser that we used was "stat_parser" implemented by (Emil Emelio, 2014). Once the trained parser was ready, we tested the parser functionality and visualised the parsed sentences to analyse them. After analyses, the hand written rules were specified based on patterns analysed through the parsed sentences.

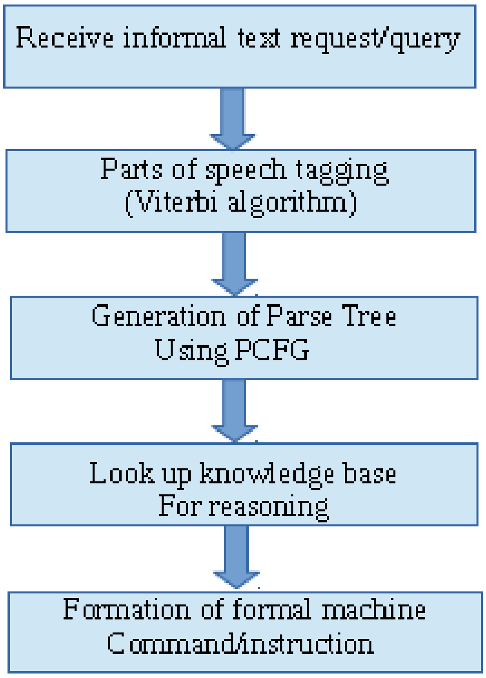

Receive informal text request/query
Parts of speech tagging (Viterbi algorithm)
Generation of Parse Tree Using PCFG
Look up knowledge base For reasoning
Formation of formal machine Command/instruction

Fig.4. Processing Pipeline

Once the object is extracted, we analyse if there is a direct entry for that object in the knowledge base. If there exists a direct entry in the knowledge base, we look up whatever information the query is asking for. If the knowledge base lacks a direct entry for the extracted object, it implies that the object is specified as a possessive proposition attached to a noun phrase. For instance, in the sentence, “Where is my mom”, the object is “my mom”. This object does not contain a direct entry in the knowledge base for “my mom”. Hence, a few more steps are required to extract a proper noun for which an entry in the knowledge base exists. To that end, all familial relations are maintained in a separated values file. The csv file is parsed to extract the name of the person in the object part of the sentence and then the required info is fetched from the knowledge base. After that information is fetched, as per the command or query proper actions are performed. Like stated above in an example like “where is my daughter?” The speaker is identified by its mobile Bluetooth mac address and then the system looks up in relation table to find who the daughter of the speaker is. After identification of the daughter, the location tracking module comes into picture. This module uses Bluetooth triangulation technique to get the exact location of the person. The main step here is to calculate the Received Signal Strength Indicator (RSSI) value of the Bluetooth. This is used to get a correlation to the distance between sender and receiver in a network.

The picture above is a demonstration of Bluetooth triangulation where three Bluetooth modules (Access Point i.e. AP) are placed in a room and then by using the formula we get the actual coordinates of the tracking device (D). For simplicity sake, only one device is considered. The device being tracked is connected to three Access Points (AP) whose positions are known in the XY plane. The distance between an Access Point (AP) and the device being tracked is based on the principle of motion and least square statistical method. Using these methods we estimate the location of the person. As soon as the system gets the location of the person in query, that location is sent back to the user. This location can be represented either via text notification or via voice notification.

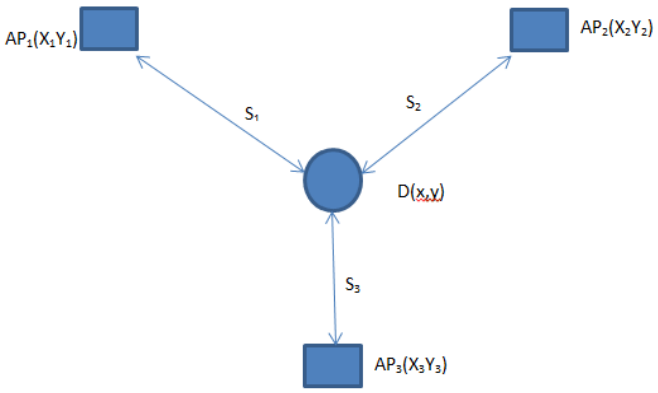


Fig.5. Bluetooth Triangulation

The algorithms used for extracting the noun phrase, object and subject are briefly outlined as follows.

```
Algo Extract Noun Phrase (Tree t)-
1. Length := number of children nodes
2. If Length <= 0 then, return None
3. If Length > 0 and node name = “NP”then, return the t.
4. Recursively call Extract Noun Phrase with right child of
   tree.
```

```
Algo Extract Object (Tree t)-
1. Length := number of children nodes
2. If Length <= 0 then, return None
3. If Length > 0 and node label = “NP”or “VP”or “PP”then,
   return the t.
4. Recursively call Extract Noun Phrase with right child of
   tree.
```

```
Algo Extract Subject(Tree t)-
1. Length := number of children nodes
2. If Length > 2 then, perform steps 3
3. If left child of tree has a label “NNS”return the leaves of
   that tree
4. Return None
```

RESULTS AND DISCUSSION

After the methodologies, the next important thing to be done is to understand the test cases where we can understand the result of our project.

Test Case 1-Location based queries.

In this case, the indoor location of the users is queried and the actual output is matched against the expected output. The user queries that follow are partitioned into two- ones that produce the accurate result and the ones that do not. The inaccurate queries and commands are listed at the end. Note that the second category is not a finite set because all the queries that could theoretically produce inaccurate results may not necessarily be relevant and grammatically correct queries.

The list of queries about location that produce the correct output is-

“Where is Anup?”,”Where is mom?”, “Where is my mum?”, “Where can I find my dad?”, “Do you know where is Kaushal?”

Test Case 2- Room usage based queries:

This type of query is the one that enquires about the room usage statistics. The queries of such type start with “How”. Similar to the previous case, the test results are divided into the ones that are successful and the ones that do not yield the accurate results. The inaccurate queries are given at the end of the test cases.

The list of queries about location that produce the correct output is-

“How many people are there in my room?”, “How many rooms are in use?”, “How many people are there in mums room?”
“How many rooms are empty?”,” How many people are there in the house?”

Test case 3- Commands that control the lights and fans-

In this case, different command is fired by the user mainly to control the functionality of the appliances which includes the light and the fan. As written above, here also two different test results are generated i.e. the successful and unsuccessful results.

The list of queries about location that produce the correct output is-
“Turn on/off the lights”, “Turn off the lights at 6:00 pm”, ”Turn off the fan I leave the room”

Future Scope-

Talking about the future scope our system, firstly we will be overcoming the main constraint of our project i.e. throughout the project; we’re assigning unique identity to a person by associating that person to his/her mobile phones. This puts an unusual constraint in interacting with the system and has to carry the mobile phones throughout the house. To eliminate this constraint, new forms of interacting to system can be explored such as smart watch. Secondly, the thing which will be considered for future scope will be replacing the method driven approach by supervised machine learning algorithm. For this purpose two different methods are planned to be used. First, an artificial neural that would be trained on supervised data. The labeled data on which the neural network will be trained would be a two tuple system consisting of informal speech command and corresponding machine understandable instruction. Second, would be Naïve Bayes classification algorithm where the system will classify an informal statement into categories based on different parameters.

Conclusion

In conclusion, the system is designed to improve the standard of living in home. We’re mainly looking to make the home a cognizant entity that understands human language and performs the required action appropriately. A potentially direct application of such a system will be as an aid for elderly and physically challenged persons. The system will be a convenient way to use and control the home automation elements for elderly persons and physically challenged people.

REFERENCES

AdwaitRatnaparkhi, 1998. “A Maximum Entropy Model for Part-Of-Speech Tagging” in the University of Pennsylvania

Agarwal Himashu, AmniAnirudh, 2006. “Part of Speech Tagging and Chunking with Conditional Random Fields” in the proceedings of NLPAI Contest.

Antony P.J, Santhanu P Mohan, Soman K.P, 2010. ”SVM Based Part of Speech Tagger for Malayalam”, IEEE International Conference on Recent Trends in Information, Telecommunication and Computing, pp. 339-341.

Brants, TnT 2000. A statistical part-of-speech tagger. In Proc. of the 6th Applied NLP Conference, pp. 224-231.

Cutting, J. Kupiec, J. Pederson and P. Sibun, 1992. A practical partof-speech tagger. In Proc. of the 3rd Conference on Applied NLP, pp. 133-140.

D Jurafsky, 2000 JH Martin, Speech and Language Processing.

Dinesh Kumar, Gurpreet Singh Josan, 2010. ”Part of Speech Taggers for Morphologically Rich Indian Languages: A Survey”, *International Journal of Computer Applications,* Volume 6–No.5, pp. 1-9

Manish Shrivastava and Pushpak Bhattacharyya, 2008. Hindi POS Tagger Using Naive Stemming: Harnessing Morphological Information Without Extensive Linguistic Knowledge, International Conference on NLP (ICON08), Pune, India.

Michael Collins, 1999. “Head-Driven Statistical Models for Natural Language Parsing”

NidhiAdhvaryu, PremBalani, 2014. “Survey: Part-Of-Speech Tagging in NLP, *International Journal of Research in Advent Technology*,” (E-ISSN: 2321-9637)

PVS Avinesh, G Karthik, 2006. ”Part-Of-Speech Tagging and Chunking using Conditional Random Fields and Transformation Based Learning” in the proceedings of NLPAI Contest.

Roni Rosenfeld, 2000. Two decades of statistical language modeling: where do we go from here?

Stanley F. Chen, Joshua Goodman, 1996. An Empirical Study of Smoothing Techniques for Language Modeling. Proceedings of the 34th Annual Meeting of the ACL.

Sumam Mary Idicula and Peter S David, 2007. A Morphological processor for Malayalam Language, South Asia Research, *SAGE Publications*.

****