ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)
On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

On Voice Activated Information Retrieval System

S. Mala Devi[#1], B. Indhuja[#2], Dr. S. Murugavalli[#3]

[#1]Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, Tamilnadu, India.

[#2]Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, Tamilnadu, India.

[#3]Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, Tamilnadu, India.

ABSTRACT— Speech Recognition (SR) is a process that transcribes speech into text using a computer. Speech recognition system is a speech-to-text conversion wherein the output of the system displays text corresponding to the recognized speech. A step towards a more natural, "human-like" communication between machines and users in need of information is represented by the introduction of speech language technologies into Information Retrieval system. The integration of the Information Retrieval system (IR) and the Automatic speech recognition (ASR) system degrades the performance. The failure to recognize a keyword in continuous speech may drastically affect the performance of the IR system. The proposal is to build an ASR system that is trained to identify the word we pronounce in restricted domain. Then for each of the recognized word, the system is expected to find the match which is then extracted by the IR system.

KEYWORDS— Automatic Speech Recognition (ASR), Information Retrieval (IR), Word Error Rate (WER).

I. INTRODUCTION

Human beings find it easier to communicate and express their ideas via speech. In fact, using speech as a means of controlling one's surroundings has always been an intriguing concept. For this reason, automatic speech recognition (ASR) has always been a renowned area of research. Over the past decades, a lot of research has been carried out in order to create the ideal system which is able to understand continuous speech in real-time, from different speakers and in any environment. Automated question answering with the help of ASR has also been a topic of research and development since the earliest AI applications. Computing power has increased since the first such systems were developed, and the general methodology has changed from the use of hand-encoded knowledge bases about simple domains to the use of text collections as the main knowledge source over more complex domains.

The integration of the information retrieval system with the ASR system could be used in many domain specific applications for easy retrieval of the required information. Although any task that involves interfacing with a computer can potentially use ASR, the following applications are the most common right now [3].

A key aspect of any human–computer speech application is that of a *dictionary*: a list of all words in the vocabulary for that application, along with their pronunciations. As computers increase in power, there is less reason to restrain dictionary size, but it is necessarily finite.

Some domains are complex domains that have a history of users attempting to streamline the process to find specific information. An example of such a domain is that of medicine. It is important for a doctor to quickly diagnose the illness of a patient, and to determine if a patient is developing a new variation of an illness that has occurred before. Given the importance of finding the correct diagnosis and treatment, trusted resources can be used for question answering in this domain.

With the introduction of ASR [13], [14], 15] it is easy for the differently-abled to attend the examinations by mere utterance. This in turn dilutes the essentiality of a scribe. Answer could be retrieved from online e books. Also dictation is the most common use for ASR systems today. This includes medical transcriptions, legal and business dictation, as well as general word processing. In some cases special vocabularies are used to increase the accuracy of the system.

II. Automatic Speech Recognition

The general problem with ASR faced by any speaker in any environment in is still yet to be completely solved. Despite this, ASR technology is currently considered to be used in various applications that are to be used in limited domains. In order to analyze its viability, there are several criteria to be considered in the different speech recognition applications: the size of the vocabulary, the speech type, and the environment in which the speaker speaks. Speech recognition is easier if the size of the vocabulary is small. Another aspect is the speech type. Isolated word recognition, where each word is uttered with pauses in isolation, is easier than continuous speech recognition, where the boundaries of the words are not clearly defined. It can also be seen that read speech is easier than spontaneous speech, even though both are continuous speech recognition. A third aspect is the environment conditions in which the system is used. In a low-noise laboratory environment using close-talk microphones, the recognition performance is highly accurate than in a noisy environment, where a telephone is used for communication with the ASR system [1].

The evaluation of this process is usually measured by the word error rate (WER), which is defined as the average number of substitutions, insertions, and deletions of words that are needed to transform the recognized sentence into the correct transcription of the speech input. Some recognition results for different systems are: a WER of 0.5% for a task of isolated digits recognition; a WER of 3% for a task of read speech with a vocabulary of 5000 words; a WER of 10% for a task of broadcast news transcription with a vocabulary of 64 000 words; and a WER of 20% for a task of conversational telephone speech with a vocabulary of 64 000 words [8].

The ASR process consists of obtaining a sequence of words from an audio signal. The mathematical formulation of this process from a statistical point of view is as follows [2]:

$$\widehat{\mathrm{W}} = \underset{\mathrm{W}\in\Sigma^*}{\operatorname{argmax}}\, \mathrm{P}(\mathrm{W}|X)$$

Where $\widehat{W}$ is the best of all possible sentences (sequences of words over a vocabulary Σ) according to the sequence of acoustic features (X) obtained from the audio signal by means of a parameterization process.

Bayes-based confidence measure (BBCM) is a probability, which is interesting itself from the practical and theoretical point of view. If applied with word density confidence measure (WDCM), BBCM dramatically improves the discrimination ability of the false acceptance curve when compared to WDCM itself. Applying the Bayes' theorem and assuming that $P(X)$ is the same for all sentences, the maximization process can be formulated as follows:

$$\widehat{\mathrm{W}} = \underset{\mathrm{W}\in\Sigma^*}{\operatorname{argmax}} \frac{\mathrm{P}(\mathrm{X}|\mathrm{W})\mathrm{P}(\mathrm{W})}{\mathrm{P}(\mathrm{W})} = \underset{\mathrm{W}\in\Sigma^*}{\operatorname{argmax}}\, \mathrm{P}(\mathrm{X}|\mathrm{W})\, \mathrm{P}(\mathrm{W})$$

Thus, the recognition process consists of finding the sentence $\widehat{W}$ that maximizes the product of two different probabilities: the $P(X|W)$ provided by the so-called acoustic model and $P(W)$ provided by the so-called language model.

III. Information Retrieval

An information retrieval process begins when a user enters a query into the system. Queries are formal statements of information needs, for example search strings in web search engines. In information retrieval a query does not uniquely identify a single object in the collection. Instead, several objects may match the query, perhaps with different degrees of relevancy. Most IR systems compute a numeric score on how well each object in the database matches the query, and rank the objects according to this value. The top ranking objects are then shown to the user. The process may then be iterated if the user wishes to refine the query [7].

From a general viewpoint, the answer extraction strategies used by most IR systems can be grouped into three major categories [2]:

A. Collection-based strategies

They search only in a static textual collection in order to find the answer and return it with a justification passage

B. Web based strategies

They use the web to find the answer to the question and then return a justification extracted from a static collection.

C. Database Approaches

The possible answers to a question are stored in a database; the system only needs to pick up the answer and return it, eventually with a justification passage.

IV. The Proposed System

Speech, being one of the most natural ways to interact, has inspired people to develop engines that could record the diction of individuals and process them. This paved way to the development of systems that could be operated upon at the command of voice. For an ASR system, a speech signal refers to the analogue electrical representation of the acoustic wave, which is a result of the constrictions in the vocal tract.

An ASR system followed by an IR stage produces a voice search system. Voice search is the technology intended to provide users with the information they request with a spoken query [4]. The information requested often exists in a structured or unstructured large database (e.g., the Web being a huge, unstructured database). The query has to be compared with fields in the database or ''documents'' in the Web to obtain the relevant information. This application could be used in many fields especially for physically challenged people like blind students. They could prepare for their exams online by retrieving the information from the e- books available on the internet.

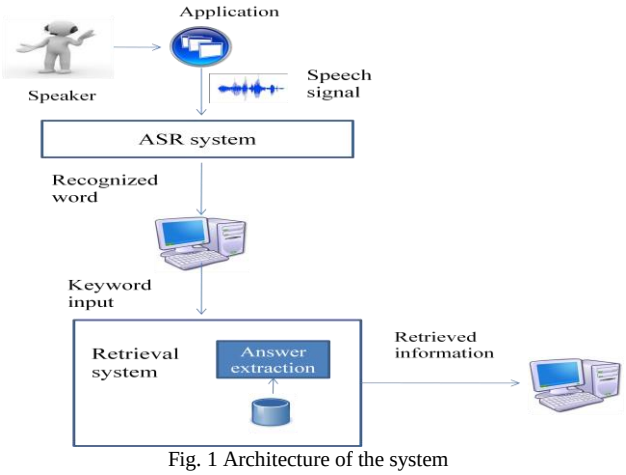

Fig. 1 Architecture of the system

The proposal is to build an ASR system that is trained to identify the word we pronounce in restricted domain. Then for each of the recognized word, the system is expected to find the word that matches. It is then extracted from the database present in the IR system.

A. ASR System

It consists of the following components:

1) *Data Acquisition:* It performs the task of acquiring human voice, converting to proper format and storing the data.

2) *Feature Extraction:*The feature extraction stage extracts a number of predefined features from the processed speech signal. These extracted features must be able to discriminate between classes while being robust to any external conditions, such as noise.

The acoustical parameters of spoken signal used in recognition tasks have been popularly studied and investigated, and being able to be categorized into two types of processing domain: First group is spectral based parameters and another is dynamic time series. The most popular spectral based parameter used in recognition approach is the Mel Frequency Cepstral Coefficients called MFCC [11], [5], [6]. Due to its advantage of less complexity in implementation of feature extraction algorithm, only sixteen coefficients of MFCC corresponding to the Mel scale frequencies of speech Cepstrum are extracted from spoken word samples in database. All extracted MFCC samples are then statistically analyzed for principal components, at least two dimensions minimally required in further recognition performance evaluation.

3) *Linguist:* The *Linguist* generates the SearchGraph that is used by the decoder during the search, while at the same time hiding the complexities involved in generating this graph. The Linguist is a pluggable module, allowing people to dynamically configure the system with different Linguist implementations. A typical Linguist implementation constructs the SearchGraph using the language structure as represented by a given LanguageModel and the topological structure of the AcousticModel (HMMs for the basic sound units used by the system).The Linguist may also use a Dictionary (typically a pronunciation lexicon) to map words from the LanguageModel into sequences of AcousticModel elements. When generating the SearchGraph, the Linguist may also incorporate sub-word units with contexts of arbitrary length, if provided. The Linguist itself consists of three pluggable components: the LanguageModel, the Dictionary, and the AcousticModel, which are described in the following sections.

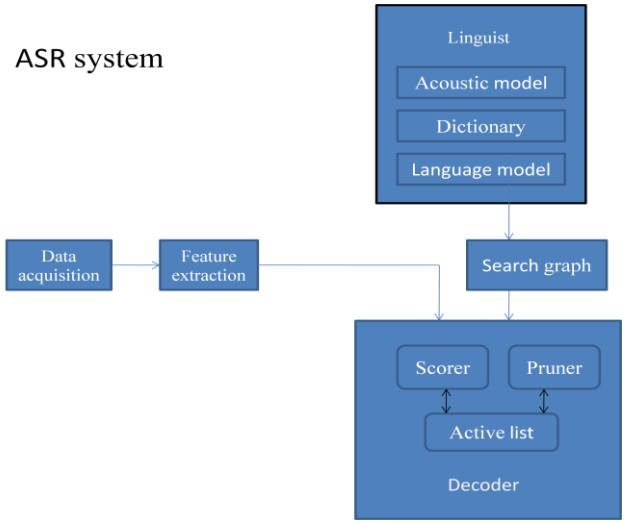

Fig. 2 ASR system

LanguageModel

The LanguageModel module of the Linguist provides word-level language structure, which can be represented by any number of pluggable implementations. These implementations typically fall into one of two categories: graph-driven grammars and stochastic N-Gram models. The graph-driven grammar represents a directed word graph where each node represents a single word and each arc represents the probability of a word transition taking place. The stochastic N-Gram models provide probabilities for words given the observation of the previous n-1 words.

Dictionary

The *Dictionary* provides pronunciations for words found in the LanguageModel. The pronunciations break words into sequences of sub-word units found in the AcousticModel. The Dictionary interface also supports the classification of words and allows for a single word to be in multiple classes [9].

Acoustic Model

The *AcousticModel* module provides a mapping between a unit of speech and an HMM that can be scored against incoming features provided by the Frontend. As with other systems, the mapping may also take contextual and word position information into account. Typically,

the Linguist breaks each word in the active vocabulary into a sequence of context-dependent sub-word units. The Linguist then passes the units and their contexts to the AcousticModel, retrieving the HMM graphs associated with those units. It then uses these HMM graphs in conjunction with the LanguageModel to construct the Search Graph. In this graph, each node corresponds to an HMM state and each arc represents the probability of transitioning from one state to another in the HMM. Furthermore, the number of states in an HMM may vary from one unit to another in the same AcousticModel. Each HMM state is capable of producing a score from an observed feature. The actual code for computing the score is done by the HMM state itself, thus hiding its implementation from the rest of the system, even permitting differing probability density functions to be used per HMM state..

SearchGraph

Even though Linguists may be implemented in very different ways and the topologies of the search spaces generated by these Linguists can vary greatly, the search spaces are all represented as a SearchGraph. Illustrated by example in Fig. 3, the SearchGraph is the primary data structure used during the decoding process.

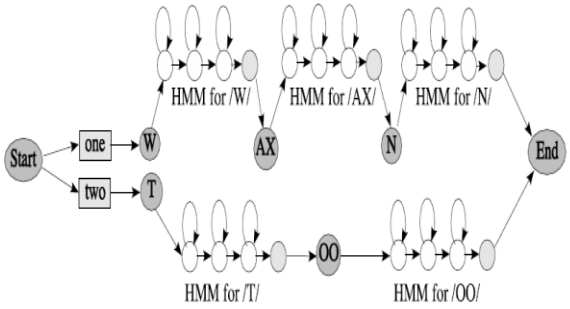

Fig. 3 Example for a Search graph

The graph is a directed graph in which each node, called a *SearchState*, represents either an *emitting* or a *non-emitting* state. Emitting states can be scored against incoming acoustic features while non-emitting states are generally used to represent higher-level linguistic constructs such as words and phonemes that are not directly scored against the incoming features. The arcs between states represent the possible state transitions, each of which has a probability representing the likelihood of transitioning along the arc. The Search Graph interface is purposely generic to allow for a wide range of implementation choices, relieving the assumptions and hard-wired constraints found in previous recognition systems. In particular, the Linguist places no inherent restrictions on the following:

- Overall search space topology
- Phonetic context size
- Type of grammar (stochastic or rule based)
- N-Gram language model depth

A key feature of the Search Graph is that the implementation of the Search State need not be fixed. As such, each Linguist implementation typically provides its own concrete implementation of the Search State that can vary based upon the characteristics of the particular Linguist.

4) Decoder: The primary role of the Decoder block is to use Features from the FrontEnd in conjunction with the SearchGraph from the Linguist to generate *Result* hypotheses. The Decoder block comprises a pluggable *SearchManager*. The Decoder merely tells the SearchManager to recognize a set of Feature frames. At each step of the process, the SearchManager creates a *Result* object that contains all the paths that have reached a final non-emitting state. To process the result, system also provides utilities capable of producing a lattice and confidence scores from the Result. Unlike other systems, however, applications can modify the search space and the Result object in between steps, permitting the application to become a partner in the recognition process. Like the Linguist, the SearchManager is not restricted to any particular implementation. Each SearchManager implementation uses a token passing algorithm. A token is an object that is associated with a SearchState and contains the overall acoustic and language scores of the path at a given point, a reference to the SearchState, a reference to an input Feature frame, and other relevant information. The SearchState reference allows the SearchManager to relate a token to its state output distribution, context-dependent phonetic unit, pronunciation, word, and grammar state. Every partial hypothesis terminates in an active token. As it is a common technique, however, the system provides a sub-framework to support SearchManagers composed of an *ActiveList*, a *Pruner* and a *Scorer*.

Active list:

The implementations of a SearchManager may construct a set of active tokens in the form of an *ActiveList* at each time step.

Pruner:

The SearchManager sub-framework generates ActiveLists from currently active tokens in the search trellis by pruning using a pluggable *Pruner*. Applications can configure implementations of the Pruner to perform both relative and absolute beam pruning. The implementation of the Pruner is greatly simplified by the garbage collector of the Java platform. With garbage collection, the Pruner can prune a complete path by merely removing the terminal token of the path from the ActiveList. The act of removing the terminal token identifies the token and any unshared tokens for that path as unused, allowing the garbage collector to reclaim the associated memory.

Scorer

The SearchManager sub-framework also communicates with the *Scorer*, a pluggable state

probability estimation module that provides state output density values on demand. When the SearchManager requests a score for a given state at a given time, the Scorer accesses the feature vector for that time and performs the mathematical operations to compute the score. The Scorer retains all information pertaining to the state output densities. Thus, the SearchManager need not know whether the scoring is done with continuous, semi-continuous or discrete HMMs. Furthermore, the probability density function of each HMM state is isolated in the same fashion.

B. Retrieval System

Due to the semantic disconnect between query and documents, IR is liable to return a lot of irrelevant data. So, the IR System has to interpret and rank its documents, according to how relevant or how exactly do they relate to the user's query. The main aim of information retrieval system is relevance. Hence the IR system condenses and simplifies searchable documents by getting a logical view of each doc. A set of keywords ("index terms") that are representative of the document is first formulated. Store these signatures for a set of documents together in one small and quickly searchable file. To keep the size of this file small, the stop words ("and", "the") are filtered, and words are reduced to their roots ('clone' from cloning or cloned). Also the list can be limited to a list to nouns, and thus the list can be compressed the list. Now we have a neat, easily searchable index for these documents. The keyword with the help of which the search is made comes from the ASR system in the form of recognized word with the help of which the required information is retrieved.

V. RESULTS

Unlike most speech recognition systems, which represent the HMM graphs as a fixed structure in memory, here HMM is merely a directed graph of objects. By representing the HMM as a directed graph objects instead of a fixed structure, an implementation of the AcousticModel can easily supply HMMs with different topologies. The AcousticModel also allows sharing of various components at all levels. That is, the components that make up a particular HMM state such as Gaussian mixtures, transition matrices, and mixture weights can be shared by any of the HMM states to a very fine degree .The modular framework permits to do some things very easily that have been traditionally difficult. For example, both the parallel and Bushderby SearchManager implementations were created in a relatively short period of time and did not require modification to the other components of the system.

Also it provides with the ability to use modules whose implementations range from general to specific applications of an algorithm. For example, it would be able to improve the runtime speed for the RM1 regression test by almost 2 orders of magnitude merely by plugging in a new Linguist and leaving the rest of the system the same. WER is comparatively low. The result of the training obtained is

MODULE: DECODE Decoding using models previously trained (2014-02-01 12:49)

```
        Decoding 130 segments starting at 0
        (part 1 of 1)
        pocketsphinx_batch Log File          completed

Aligning results to find error rate

        SENTENCE ERROR: 69.2% (90/130) WORD
        ERROR RATE: 24.4% (188/773)
```

VI. CONCLUSION

An ASR system for recognizing the word pronounced is developed and the related information could be retrieved from the database. HMM based models are producing better recognition results. Although progress has been made, full understanding of language is not reached, even with current human language technology. Therefore, voice-activated retrieval systems currently do not have the "human-like" communication that we would like to have in a machine. Nevertheless, voice-activated IR systems can indeed be useful, specifically for restricted-domain IR tasks, where more limited knowledge is needed. The next generation of voiced-activated IR systems will allow users to have a deeper speech interaction in order to obtain answers from different kinds of repositories of unstructured information (e.g., web pages, newspapers, and books in a digital format). Nowadays, this ambitious goal can be attempted with the help of advances in the principal fields involved in these processes: speech processing and IR/QA systems. However, there are important and specific difficulties that still make this task a challenge.

REFERENCES

[1] Paolo Rosso, Llu´ıs-F. Hurtado and Encarna Segarra,Emilio Sanchis, *"On the Voice-Activated Question Answering", IEEE Transactions on Systems, Man, and Cybernetics—part c: applications and reviews, vol. 42, no. 1,January 2012 75.*

[2] L. Bahl, F. Jelinek, and R. Mercer, "*A maximum likelihood approach to continuous speech recognition," IEEE J. Pattern Anal. Mach. Intell.*, vol. PAMI-5, no. 2, pp. 179–190, Mar. 1983.

[3] Wiqas Ghai and Navdeep Singh, *"Literature Review on Automatic Speech Recognition, "*International Journal of Computer Applications (0975 – 8887) Volume 41– No.8, March 2012.

[4] Y. Wang, D. Yu, Y. Ju and A. Acero, ''*An introduction to voice search,'' IEEE Signal Process. Mag., vol. 25, no. 3, pp. 28–38, May 2008.*

[5] Niladri Sekhar Dey,Ramakanta Mohanty, K.L.chugh, *"Speech and Speaker Recognition System using Artificial Neural Networks and Hidden Markov Model",2012 International Conference on Communication Systems and Network Technologies.*

[6] Santosh K.Gaikwad, Bharti W.Gawali and Pravin Yannawar, *"A Review on Speech Recognition Technique," International Journal of Computer Applications | Vol. 10-No.3, November 2010.*

[7] NIST. Speech recognition scoring package (score). [Online]. Available: http://www.nist.gov/speech/tools

[8] S. J. Young, N. H. Russell, and J. H. S. Russell, *"Token passing: A simple conceptual model for connected speech recognition systems,"* Cambridge University Engineering Dept, UK, Tech. Rep. CUED/F-INFENG/TR38, 1989.

[9] Gerhard Weikum, Gjergji Kas neci, Maya Ramanath, and Fab ian Suchanek, *"Database and Information-Retrieval Methods for Knowledge Discovery,"* Communications of the ACM | april 2009 | vol. 52 | no. 4.

[10] Chadawan Ittichaichareon, Siwat Suksri and Thaweesak Yingthawornsuk, *"Speech Recognition using MFCC," International Conference on Computer Graphics, Simulation and Modeling (ICGSM'2012) July 28-29, 2012 Pattaya (Thailand)*

[11] Douglas O'Shaughnessy, *"Interacting With Computers by Voice: Automatic Speech Recognition and Synthesis," PROCEEDINGS OF THE IEEE, VOL. 91, NO. 9, SEPTEMBER 2003*

[12] F. Jelinek, "*Continuous speech recognition by statistical methods," Proc. IEEE*, vol. 64, pp. 532–556, Apr. 1976.

[13] L. Rabiner and B. Juang, "*Fundamentals of Speech Recognition"*, Englewood Cliffs, NJ: Prentice-Hall, 1993.

[14] R. Reddy, "Speech recognition by machine: A review," *Proc. IEEE*, vol. 64, pp. 501–531, Apr. 1976.

[15] Richard M. Stern and Nelson Morgan, "Hearing is believing," *IEEE signal processing magazine, November 2012*.

[16] Michelle Cutajar, Edward Gatt, Ivan Grech, Owen Casha and Joseph Micallef, *"Comparative study of automatic speech recognition techniques*", *IET Signal Process., 2013, Vol. 7, Iss. 1, pp. 25–46 25*.

[17] Hung-yi Lee, Chia-ping Chen, and Lin-shan Lee, *"Integrating Recognition and Retrieval With Relevance Feedback for Spoken Term Detection," IEEE transactions on Audio, Speech and Language processing, 2012.*

[18] Xiaodong He and Li Deng, *"Speech-Centric Information Processing: An Optimization-Oriented Approach*," *Proceedings of the IEEE | Vol. 101, No. 5,May 2013.*

[19] Haizhou Li, Bin Ma, and Kong Aik Lee, "*Spoken Language Recognition: From Fundamentals to Practice," Proceedings of the IEEE | Vol. 101, No. 5, May 2013*