Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 1, January 2014, pg.501 – 507

RESEARCH ARTICLE

Speech Recognition Using Backoff N-Gram Modelling in Android Application

S.Aparna[1], V.Senthil Kumar[2]

[1]PG Student, M.E Computer and Communication & Anna University

[2]Associate Professor, Electronics and Communication &Anna University

Ganadipathy Tulsi's Jain Engineering College, Vellore, TamilNadu

[1] aparna.prabha1@gmail.com; [2] kvcs2000@gmail.com

Abstract— Google is one of the most popular information retrieval systems among users. Spoken questions are a natural standard for penetrating the network in settings where typing on a console is not applicable. This paper describes a speech boundary to the Google search. The study entails the improvement of Hands-Free voice recognition Google Search Engine to operate Google and browse the result of search without using a keyboard or mouse. Speech recognition uses are becoming more and more beneficial nowadays. Digital processing of speech signal and voice recognition process is very important for fast and precise automatic voice recognition technology. Here we present a new service which is not currently accessible in Google search engine (GSE). It suggests the enactment of speech recognition input in GSE. The paper stimulates an older method from n-gram language modelling to scale training data. The algorithm is implemented efficiently using a MapReduce/SS (Spectral Subtraction) Table framework based on HMM and Gaussian Mixture models.

Keywords— Automatic speech recognition; acoustic modelling; Hidden Markov models; phonetic context; back-off; n-gram; distributed storage.

I. INTRODUCTION

Automatic speech recognition (ASR) has received substantial attention and has achieved remarkable performance in noise-free environments. However, under more sensible conditions where background, additive and convolutional noise is present; performance degrades significantly, discouraging its practical use[3.]The literature on robust ASR deliberates numerous approaches to handle this problem. Some approaches achieve robustness using one, or a sequence of techniques that can be grouped as speech enrichment or pre-processing techniques, robust feature extraction methods, feature post processing techniques, and model adaptation to noisy environments. Spectral subtraction (SS) methods[9]belong to the class of speech enhancement techniques that have been largely applied in ASR contexts. However, most speech improvement methods are personalized at improving speech intelligibility for human listeners and hence, they may not carry out well in ASR tasks[4]. These methods aim at improving the worth of the noisy speech by reducing the noise while minimizing the speech alteration introduced during the enhancement process. There are basically three causes of errors when spectral subtraction is applied to noisy speech, such as magnitude, cross-term and phase errors.

As web-centric computing has developed over the last period, there has been an increase in the volume of data available for training aural and language standards in speech perception. Machine conversion and language modelling for Google Voice Search[3] have shown that using more training data is quite beneficial for

improving the performance of mathematical language models. In a number of language processing tasks; we cope with the problem of recovering a series of English words after it has been distorted by means of access through a noisy channel. To attempt this problem successfully, we must be able to estimate the possibility with which any specific sequence of English words will be turned-out as input to the noisy channel[1]. In this paper we discuss a method for making such estimates. Ngram language models are abundant in automatic speech recognition (ASR).

II. BACK-OFF NGRAM MODEL

Back-off is a generative n-gram language model that evaluates the provisional possibility of a word given its past record in the n-gram[1].It achieves this estimation by "backing-off" to standards with smaller histories under specific restrictions. With such act, the model with the most dependable information about a given history is used to produce enhanced results. An n-gram model can be easily calculated by adding the number of conversions from history 'h' to a term't'. This method is known as Maximum Likelihood Estimation (MLE). However, unless we have sufficiently large training collection which makes the ML estimate definite, our model may be embarrassing if queries ht has never appeared in the training corpus, the case is denoted by C(ht)=0.C(ht)=0 is embarrassing because "unseen" does not mean "impossible", that we do not see a transition from h to t in a collection does not mean that this transition will not happen.

A common solution to this embarrassing is "smoothing" over the function of w, P(t|h) for every h, and make those t's with P(t|h)=0 (i.e., C(ht)=0) a value little larger than 0. An easy and well known method is Laplacian smoothing, does not toll any wealthy people; as a substitute, it gives every people rich and poor a cent. There are many ways to distribute the toll from rich to the poor. Backoff is one of them. If C(ht)=0, the collected toll is given to P(t|h) by taking C(h't) into deliberation. Back-off a non-linear model, the estimate of n-gram is allowed to back off via increasingly shorter records. It is the brief model that can produce sufficient information about the current setting used. The problem is probability estimates can vary rapidly on adding more data when back-off algorithm selects a unlike order of n-gram model.

.

III. EXISTING MODEL

Recent applications have led to availability of data far beyond that commonly used in ASR systems. Filtering utterances logged by the Google Voice Search service at an adequate ASR confidence threshold, guarantees transcriptions that are close to human annotator performance[1]. A similar approach for automatic speech recognition (ASR) acoustic modelling that is conceptually simpler than established techniques, but more aggressive in this respect.

The most common technique for dealing with data sparsity when estimating context-dependent output distributions for HMM states is the well-known decision-tree (DT) clustering approach[6]. The back-off acoustic model is estimated, stored and served using MapReduce distributed computing infrastructure. Speech recognition experiments are carried out in an N-best list rescoring framework for Google Voice Search. The clustered states have enough data for reliable estimation; the algorithm guarantees a minimum number of frames at each context-dependent state (leaf of the DT).

IV. MODELLING IN DISTRIBUTED ENVIRONMENT

BAM evaluation and run-time are implemented using MapReduce and SSTable, drawn heavily from the large language modelling approach for statistical machine translation described in Large Machine Models in Machine Translation.[2]

A. Estimation Using MapReduce

MapReduce is a software design and an associated execution for processing and making enormous data sets. Users state a map function that processes a key/value pair to produce a group of altering key/value pairs, and a reduce function that combines every intermediary values associated with the identical transitional key[2]. Many tangible tasks are expressible in this model, as shown in the paper. We gathered that most of our calculations occupied applying a map operation to each logical profile in our input in order to compute a group of transitional key/value pairs, and then use a reduce operation to all the values that shared the same key, in order to combine the secondary data suitably. Our use of a functional model with user specified map and reduce activity allows us to parallelize large computations without obstacles and to use re-execution as the primary procedure for error tolerance. The foremost role of this work is a simple and effective interface that enables mechanical parallelization and allocation of large-scale computations, combined with an application of this boundary that achieves extreme performance[8].

The process takes a group of input key/value pairs, and generates another group of output key/value pairs. The user of MapReduce expresses the estimation as two functions: Map and Reduce. Map function built by the user, takes an input pair and produces a group of intermediate key/value pairs. The MapReduce collection groups all intermediate values associated with the equivalent intermediary key and elapses them to the Reduce

function. The Reduce function, built by the user, takes a transitional key and a group of values for that key. It combines together these values to form a possibly smaller group of values. Naturally just zero or one output value is generated for each Reduce supplication. The transitional values are provided to the users reduce function by an iterator.

B. BAM Test using Spectral Subtraction

Speech augmentation aims to improve speech quality by using various algorithms. The goal of augmentation is improvement in perspicuity and/or overall mental quality of corrupted speech signal using aural signal processing methods. Improving of speech degraded by noise, is the most significant field of speech enhancement and used for many applications; some of them are cellular phones, teleconferencing, and speech identification. The process of speech augmentation for noise diminution are categorized into fundamental classes, this includes filtering techniques such as spectral subtraction method [1]. Spectral Subtraction (SS) is an algorithm formed to reduce the worsening effects of noise acoustically added in speech signals [9].

C. Parameter Estimation using Maximum Likelihood Approach

Accurate acoustic modelling contributes greatly to improving speech recognition performance. The conventional structure for accurate modelling is based on hidden Markov models (HMMs)[7] and Gaussian mixture models (GMMs) where the mathematical model parameters are obtained by the maximum likelihood approach. The conventional ML-based framework presents problems for real speech recognition tasks because it cannot select an appropriate model structure in principle, and often estimates incorrect parameters when the amounts of training data assigned for the parameters are small (over-training), which in turn degrades the perception performance.

D. Node Splitting using Decision Tree Model

Decision tree state tying based acoustic modelling has become increasingly popular for modelling speech variations in large vocabulary speech recognition[1].In this approach, the acoustic phonetic acquaintance of the objective language can be effectively integrated in the method according to a coherent maximum likelihood structure. The numerical framework of decision tree in acoustic modelling provides two major benefits over the prior rule or bottom up based methods. First, the classification and prediction power of the decision tree allows to synthesize model units or contexts, which do not occur in the training data. Second, the node splitting procedure of decision tree based state tying is a model selection process. It provides a way of maintaining the balance between model complexity and the number of parameters in order to render robust model parameter estimation from the limited amount of training data.

Recently, there are many attempts to improve the phonetic decision tree state tying based approach in acoustic modelling. Two problems in decision tree state tying are of particular interest. One is the tree growing and node splitting problem and it concerns the issue of how to find an optimal node split, given the particular parametric form of the impurity function (e.g., the likelihood of the training data). Another one is the parametric modelling problem of the data distributions during the process of node splitting [8]. For phonetic decision tree approach in aural modelling, these two problems are closely related.

The problem of ideal node ripping is about finding the prime node split, and the parametric modelling is an obstacle of stipulating a suitable metric, which defines the quality of the split. In general, construction of a globally optimal decision tree is a computationally intractable problem. The parametric forms of distributions used in decision tree node splitting are based on Gaussian distributions. In this paper, we deliberate methods for increasing the robustness and accuracy in decision tree grouping based acoustic modelling.

E. Estimation with N-gram

The design of word estimation with probabilistic methods is called N-grams, which guess the subsequent word from the former N-1 words. Such mathematical methods of word series are called Language Models. Computing the possibility of the succeeding word will turn out to be closely related to computing the possibility of a string of words[2]. Estimators like N-grams that allocate a conditional possibility to possible subsequent words can be used to allocate a joint probability to a complete sentence. Whether estimating possibilities of succeeding words or of whole sequences, the N-gram model is one of the most needed tools in speech and language processing. N-grams are important in any assignment in which we have to classify words in noisy, indefinite input. In speech recognition, the input speeches sounds are very perplex and many words sound alike. N-gram models are also important in mathematical machine conversion.

V. SPEECH RECOGNITION

In this paper, a Hidden Markov Model (HMM) speech recognition system is designed. The ultimate aim of speech perception is to make machine understand natural language. In provisions of HMM recognition, the typical Viterbi process has been improved and recognition speed has been increased. Speech recognition is the

capability of a device or a database to receive and decipher transcription[7], or to comprehend and carry out spoken rules. For use with processor, analog audial must be reformed into digital signals. This needs an analog-to-digital translator. To interpret the signal, it must have a digital catalog and a speedy means of matching up the information with signals. The speech models are piled on the hard drive and burdened into memory when the program is run. The comparator tests these stored models with the output of the Analog-to-Digital converter. A speech perception system uses a language model to find out the possibility of different perception hypotheses. A speech-to-text system can also increase system accessibility by providing data entry options for physically challenged users.

Google has collected an enormous catalog of words derived from the regular entries in the Google search engines. The record contains more than 230 billion words[10]. If we utilize this type of speech identifier, it is likely that our voice is stored on Google servers. This circumstance stipulates constant increase of information used for training, improving accuracy of the technique. The working of speech recognition systems is usually estimated in terms of accuracy and speed. Speech is deformed by contextual sound and reverberation. Both aural modelling and speech modelling are essential parts of current mathematical based speech recognition procedures. Hidden Markov models (HMMs) are extensively used in many systems.

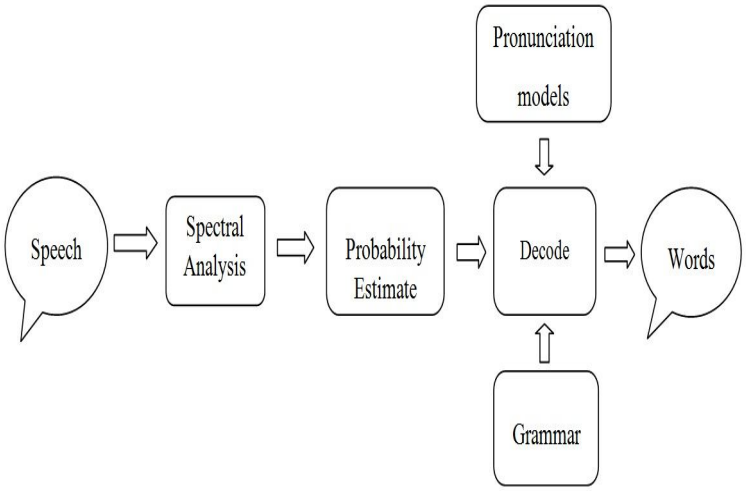

Fig 1: Process in Speech Recognition

Fig 1 represents the process in Speech conversion. Different stages are involved in speech renovation; this includes Speech Acquisition, Speech Pre-processing, Voice Activity Detection, Pre-emphasis, Probability Estimation and Decoding. Speech acquisition is the first and main step in speech recognition. It is the activity of obtaining fresh speech signals from the operator. The attained speech needs to be pre-processed before taking out information. The most essential stage in pre-processing is speech detection in the acquired signal. Speech pause detection algorithm is used to sense silence segments of a speech signal, eradicate them. Then the Human Voice is transformed to digital form. Voice activity detection is an approach to identify the un-silenced part of the incoming speech signal. This involves two steps, If the feature of the input frame exceed the estimated threshold level, verdict VAD = 1 is computed which concludes that speech is present. Otherwise, a verdict VAD = 0 is computed which states the absence of spoken words in the input frame. Finally Probability estimation is carried out by HMM and GMM models and decoding is performed by Viterbi Algorithm.

VI. VITERBI DECODING

A Viterbi decipherer uses the Viterbi process for translating a bit stream that has been encrypted using a convolutional code. There are other procedures for decoding a convolution ally encoded stream. This system is the extreme resource consuming method, also does maximum likelihood decoding[4]. It is often used for decoding convolutional codes with restraint lengths. It is an active programming system for finding the most likely series of hidden states known as Viterbi path that results in a string of empirical events mainly in the framework of Markov data basis and hidden Markov models. In speech-to-text speech detection, the aural signal is treated as the experiential series of events; the chain of text is pondered to be the hidden cause of the aural signal.

VII. PROBABLISTIC MODELS

A. Hidden Markov Model

A hidden Markov model is a numerical Markov model, here the method being is assumed to be a Markov process with unobserved states[7]. In Markov models, the state is completely visible to the viewer, therefore the state conversion probabilities are the only parameters. In hidden Markov model, the state is not promptly evident but the output dependent on the state is obvious. Every state has likelihood dissemination over the probable output symbols. Hence the series of symbols produced by a HMM gives some data about the string of states. A hidden Markov model can be treated as a simplification of a mixture model where the latent variables which control the mixture constituent to be choosed for each inspection, are related through a Markov process[5]. The Markov Model has a finite set of states; each of them is related with the probability distribution. Conversions among the states are regulated by a set of possibilities called transition probabilities. In a specific state an outcome can be produced according to the related probability distribution. It is only the consequence, the state is not visible to an external observer and therefore the states are ``hidden'' outside; hence the model is termed as Hidden Markov Model.

B. Gaussian Mixture Model

By definition a mixture model can be determined as a probabilistic model for denoting the occurrence of sub populations within the whole and overall population, without demanding that an empirical record set should detect the sub population to which a separate observation belongs[2]. Strictly a mixture model represents the probability distribution of observations in the total population. "Mixture models" are used to make mathematical deduction about the features of the sub-populations pooled, without sub-population recognizes information by Gaussian Mixture Models (GMMs)[1]. GMM is the most statistically mature methods for clustering. In the process of Clustering we assume that single data points are produced by initially picking one of the sequences of multivariate Gaussians and then selecting from them. It can be also said as a well-defined computational operation. Here we use an optimization method called as Expectation Maximization (EM).

C. Expectation Maximization Method

An expectation–maximization (EM) procedure is a repetitive method for finding maximum likelihood estimates of parameters in mathematical models, where the model depends on unobserved hidden variables[5]. The EM iteration fluctuates between performing an expectation (E) step, which generates an operation for the anticipation of the log likelihood estimated using the current estimate for the factors and a maximization (M) step, which calculates factors maximizing the expected log-likelihood found on the E step. These factor estimates are then used to regulate the allocation of the hidden variables in the next E step. The EM method is used to find the maximum likelihood factors of a mathematical model where the equations cannot be solved directly[2]. These models involve hidden variables in addition to indefinite parameters and known data observations, either there are missing values among the data else the model can be originated more simply by simulating the presence of other unobserved data points.

VIII. PROPOSED MODEL

Google Voice Search is one of the finest identifier accessible for Android, supports many languages. This service compels internet connection as Voice recognition occurs on Google servers[10]. This application has a simple action that tells users to speak. The instant user finishes talking, the recognizer detects speech. The voice search feature on Google uses voice recognition technique such that users can search based on voice generated queries. A small button on the search box can be clicked, which captures voice entry. Subsequently a quick processing cycle, the voice is converted to text and the search is executed which also displays similar words to the spoken text. The primary challenge is to design a new application from the Google Voice Search that enables Speech recognition with higher Speed, Accuracy and Robustness. There are many steps in creating an application. They are,

- Receive a request for voice recognition,
- Check the availability of application for speech recognizing,
- If speech recognizing is available, then request the intent for it and receive the results,
- If speech recognizing is not available, install the application from the apk manager.

The procedure for constructing the language model is as follows:

- Acquire queries by extracting a sample from Google's query logs.
- Separate out non-English queries by removing queries according to the language chosen.

- Use Google's spelling correction mechanism to correct misspelled queries.
- Create lists of collocations.
- Create the vocabulary consisting of the most recurrent words and collocations.
- Use a dictionary and an automatic text-to phonemes device to obtain phonetic records for the vocabulary, applying a separate method to special terms.
- Estimate n-gram probabilities to create the language model.

Here we had a created an additional database that displays similar words and sentences to the text that is spoken by the user. This includes numbers also (0-9). Consider for example the word spoken is "Hello", the result obtained is shown in table below:

Hi
Hi Hello
High Hello
Hot Hello

Table 1: Result for text "Hello"

Even numbers can be displayed as results, for example "India got its independence in 1947", and the output is shown below:

India got its independence
India got independence in 1947
India got its independence in 1947
Independence in 1947

Table 2: Result for text with numbers

By creating this additional database the confusions among Homophones in words is reduced. Perplexity is also eliminated among text, in noisy environments this serves as a useful Speech Recognition Engine.

IX. CONCLUSION AND FUTURE WORK

Thus back-off Ngram modelling is implemented in speech recognition and better results are obtained. Accuracy, Robustness and Speed are greatly improved. Probabilistic estimation also plays an important role in machine conversion and Perplexity among words or sentences. Original samples are taken by creating an additional database; Parallelism is produced by MapReduce technique. More Efficiency can be obtained by implementing training sets of both Male and Female Speakers in Noisy Environments. Future work can be enhanced by making call, texting messages through Speech. This can be done by using the same Speech to text process in these applications.

REFERENCES

[1] Ciprian chelba, Peng Xu, Fernando Pereria, and Thomas Richardson, "Large Scale Distributed Acoustic Modeling with Back-Off -Grams" IEEE Transactions on Audio, Speech and Language Processing, Vol 21,No.6,June 2013.

[2] Ciprian Chelba, Peng Xu, Fernando Pereira, "Distributed Acoustic Modeling With Back-Off N-Grams", IEEE Transactions on audio and speech processing.vol 26,2012.

[3] Hamzeh Albool, Maytham Ali, Hamdi Ibrahim, and Adib M.Monzer Habbal, "Hands-Free Searching Using Google Voice", International Journal of Computer Science and Network Security, VOL.10 No.8, August 2010.

[4] R.Sandanalakshmi, V.Martina Monfort, G.Nandhini, "A Novel Speech to Text Converter System for Mobile Applications", International Journal of Computer Applications (0975 – 8887) Volume 73– No.19, July 2013.

[5] Shinji Watanabe, IEEE Syst., "Variational Bayesian Estimation and Clustering for Speech Recognition" vol-12 No.4, 2004.

[6] Wolfgang Reichl and Wu Chou,"Robust Decision Tree State Tying for Continuous Speech Recognition", IEEE Transactions on Audio and Speech Processing" vol-8 No.5,2000.

[7] M.Gales,"Semi-tied covariance matrices for hidden markov models,"IEEE Transactions on Speech and Audio Processing", vol.7, no.3, pp.272-281, May 1999.

[8] Jimmy Lin and Chris Dyer, "Data-Intensive Text Processing with MapReduce" Manuscript prepared April 11, 2010.

[9] J. Cardenas, E. Castillo and J. Meng, "Noise-robust speech recognition system based on power spectral subtraction with a geometric approach" Acoustics 2012 Nantes Conference 23-27 April 2012.
[10] Xin Lei, Andrew Senior, Alexander Gruenstein, Jeffrey Sorensen, "Accurate and Compact Large Vocabulary Speech Recognition on Mobile Devices" Interspeech 2013.

