Proc. of Int. Conf. on Current Trends in Eng., Science and Technology, ICCTEST

Word Prediction using LPC in Language Models

Akshatha K.V[1] and D.J Ravi[2]
[1]Vidyavardhaka College of Engineering, Mysuru,
Visveswaraya Technological University, Belagavi.
Email: akshatha.ptz@gmail.com
[2]Professor, Vidyavardhaka College of Engineering, Mysuru,
Visveswaraya Technological University, Belagavi.
Email: ravidj@vvce.ac.in

***Abstract*— InLinear Prediction Filter method to predict a part-of-speech-based word prediction model and a word based linear prediction model to accomplish word prediction tasks. In order to find a set of mathematical equations to properly describe the word prediction the model is based on partial differential equations is proposed.Linear Predictive Coding (LPC) is a technique that attempts to derive the coefficients of a filter that would produce the utterance.To predict a word the system is trained first. We are extracting the feature of the word which has to be predict during training stage. In testing stage the feature of the input speech signal is compared with trained data. When the features of the input speech signal and trained data are matched then the word is predicted. This paper focuses on combining a word based n-gram and m-POS based word model to find the best matching word.**

***Index Terms*— Parts-of-speech(POS), Linear Predictive Coding(LPC), Natural Language Processing(NLP), Language Model (LM), n-gram model**

I. INTRODUCTION

Word prediction is a system which is developed to exchange the information either by speaking or writing. This increases the message composition rate and helps the people who suffers from speech disabilities. Prediction refers to the systems that guess which letters or words are likely to follow a given segment of a text. The important task within the context of Natural Language Processing (NLP) is to predict the most suitable word and this increases the keystroke saving (KSS). In speech recognition Language models are used to recognise the word that were spoken. Thus in speech recognition problem the goal is to find both the best sequence of words and POS tags.Word prediction systems are based on statistical n-gram language modelling and more sophisticated language models are developed to improve the performance of the traditional language models.A simplemethod is the linear interpolationwhichis used to add a Part-of-Speech (POS) component to a word n-gram Model. This method reduces the perplexity of the linear combined model but does not guarantee efficient use of the different information sources. The other method was developed which is based on the Latent Maximum Entropy Principle. This extends the basic principle of maximum entropy proposed to combine a hidden dependence structure. This method generates probabilistic models which is capable of capturing all information from different sources but it limits in the estimation of the model parameters. The other method called Directed Markov Random Fields was developed, this model was used to combine a word trigram model, a probabilistic model based on context-free grammar and a probabilistic

Grenze ID: 02.ICCTEST.2017.1.174

model based on latent semantic analysis. It is not clear how to combine all this information to be used in practical systems. Thuslinear predictive method was proposed in this work so we can mathematically model the system to combine all the main elements in the natural language processing. To recognise spontaneous speech the acoustic signal is weak to find the number of word. Thus speech recognizers use a language model that cut out acoustic alternatives by considering the previous words that were recognized. To find the most likely word sequence and the speech signal is given as

$$\hat{w} = \arg\max \Pr(W|A)$$

The above equation can rewrite using Bayes' rule as

$$\hat{w} = \arg\max \frac{\Pr(A|W)\Pr(W)}{\Pr(A)}$$

Pr(A) is independent of the choice of W and we can simplify the above equation as follows

$$\hat{w} = \arg\max \Pr(A|W)\Pr(W)$$

Where Pr(A|W) is the acoustic model and Pr(W) is the language model that assigns a probability to the sequence of words W. Where W can be written explicitly as a sequence of words W_1 W_2 W_3……..W_N where N is the number of words in the sequence. The notation $W_{i,j}$ to refer to the sequence of words W_i to W_j and thus by using the definition of conditional probabilities to rewrite the equation $\Pr(W_{1,N})$ as follows.

$$\Pr(W_{1,N}) = \prod_{i=1}^{N} \Pr(Wi|W_{1,i-1})$$

The effectiveness of the estimated probability distribution is measured to measure the perplexity which is assigned to a test corpus. Perplexity is an estimation of how accurately the language model can predict the next word of a test corpus. The perplexity of a test corpus set $W_{1,N}$can be calculated as 2^H, where H is the entropy and is given as follows

$$H = -\frac{1}{N}\sum_{i=1}^{N} \log_2 \widehat{Pr}\,(w_i|w_{1,i-1})$$

II. Methodology

The word prediction system is an important task within the context of Natural language processing. To predict the word in given context the main functions are training, testing and also a utility function called recording is available to capture the sample vocabulary sets for training and testing. The training uses a routine called extracting which is used to build feature vectors and testing uses the routine called matching to generate the best match for a test word. Testing also call extracting to get the feature vector for the test word and then compares the obtained feature vector to the feature vectors that was developed during training. Thus matching determines the best match and also figure of merit for that match. Testing uses a threshold on the figures of merit obtained from matching to decide if we have a match or if we don't have enough information for prediction. The block diagram is shown in below figure.

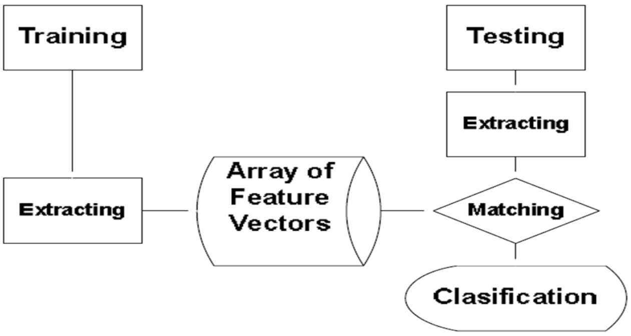

Figure 1.Block diagram

III. LANGUAGE MODELS

A. Word-based Language Model

Statistical language model is used to predict the next word based on the history of previous words. Word prediction idea was decided by probabilistic models called n-gram models which predicts the next word from the n-1 previous words. N-gram model can be viewed as unigram, bigram and trigram model.

N-Grams

N-gram is the simple model which assigns probabilities to sequences of words and sentences. If the value of N is 1 then N-gram model can be viewed as unigram. If the value of N is 2 then N-gram model can be viewed as bigram which is two-word sequence of words like "this is", "is a", "a sentence". If the value of N is 3 then N-gram model can be viewed as trigram which is three-word sequence of words like "this is a", "is a sentence". The N-gram is nothing but sequence of N word or can be viewed as predictive model that assigns a probability to next words or of whole sequences. Thus the important tools in speech and language processing is N-grammodel.

Let us study howto use N-gram models to estimate the probability of last word of an N-gram given the previous words. Let us consider an example to compute P(w | h), the probability of a word w in a given history h. Let the history h is given as "this glass is so transparent that" and we want to find the probability and the next word is 'the':

P(the | this glass is so transparent that)(1)

The probability for this sentence can be estimated from relative frequency counts: consider a large corpus and count the number of times we see'' | this glass is so transparent that'' and then count the number of times this is followed by 'the' and it is given as follows

P(the | this glass is so transparent that) = C(this glass is so transparent that the) /

C(this glass is so transparent that) (2)

Thus we can compute these counts and estimate the probability from Equation 2. Many sentences are created all the time and we cannot always be able to count entire sentences. We can also find the joint probability of an entire sequence of words by considering "this glass is so transparent" and this can be done by asking "out of all possible sequences of five words and how many of them are "this glass is so transparent?"this can be done by counting the number of time "this glass is so transparent" occurs and divide by the sum of the counts of all possible five word sequences and this takes more time to estimate. The probability of a particular random variable Xi is represented as "the", or P(Xi = "the") or we can simplify as P(the). We can represent a sequence of N words in a given sentence either as w_1.........w_n or w_1^n . In joint probability each word in a sequence can be represented as P(X = w1; Y = w2; Z = w3........W = Wn) and we can simply as P(w1; w2.....wn). How to compute probabilities of entire sequences like P(w1; w2.........wn)? This can be done by decomposing the given probability using the chain rule of probability

$$p(x_{1\ldots\ldots\ldots}x_n) = p(x_1)p(x_2|x_1)p(x_3|x_1^2)\ldots p(x_n|x_1^{n-1}) \\ = \prod_{k=1}^{n} p(x_k|x_1^{k-1}) \quad (3)$$

Now let us apply the chain rule to words and we get

$$p(w_1^n) = p(w_1)p(w_2|w_1)p(w_3|w_1^2)\ldots p(w_n|w_1^{n-1})$$

$$= \prod_{k=1}^{n} p(w_k|w_1^{k-1}) \quad (4)$$

The joint probability of a sequence and the conditional probability of a word given with previous words are linked by chain rule. From equation 4 we can estimate the joint probability of an entire sequence of words by multiplying together with conditional probabilities.Let us discuss how bigram model is used to assign the probability to word sequence in order to predict next word. Consider the below example

P(the | this glass is so transparent that) (5)

We can approximate the above equation with the probability as

P(the | that) (6)

To predict the conditional probability of the next word, we are using following approximation:

$$p(w_1|w_1^{n-1}) \approx p(w_1|w_{n-1}) \quad (7)$$

The general equation for N-gram approximation to the conditional probability of the next word in a given sequence is

$$p(w_1|w_1^{n-1}) \approx p(w_1|w_{n-N+1}^{n-1}) \quad (8)$$

Maximum likelihood estimation for the parameters of an N-gram model is obtained by getting counts from a corpus and normalize the counts so that they lie between 0 and 1. The general form of MLE N-gram parameter estimation is given as

$$p(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})} \quad (9)$$

B. POS-based Language Model

The POS tags are used to remove too much of the lexical information that is necessary for predicting the next word. It is also used to capture the syntactic role of each word as the basis of the equivalence classes. We can add POS tags into the language model by summing over all POS sequences. The speech recognition problem is to find the best word and POS sequence. Let P be a POS sequence for the word sequence W and the goal of the speech recognizer is to solve the following.

$$\widehat{\mathrm{W}}\hat{P} = \arg\max \Pr(\mathrm{W\,P}|\mathrm{A})$$

$$= \arg\max \frac{\Pr(\mathrm{A}|\mathrm{W\,P})\Pr(\mathrm{W\,P})}{\Pr(\mathrm{A})}$$

$$= \arg\max \Pr(A|W\ P)\Pr(W\ P) \quad (10)$$

III. IMPLEMENTATION

A. Pre-emphasis filtering

Speech signal has an overall spectral tilt of 5 to 12 dB per octave and a pre-emphasis filterof the form 1 – 0.99 z-1 is normally used. It is the first order filter that compensate for the fact that the lower formants contain more energy than the higher.

B. Filter

Filter is a One-dimensional digital filter.Y = filter(B,A,X) filters the data in vector X (speech signal) with the filter described by vectors A and B to create the filtered data Y. The filter is a "Direct Form II Transposed" implementation of the standard difference equation:

a(1)*y(n) = b(1)*x(n) + b(2)*x(n-1) + ... + b(nb+1)*x(n-nb) - a(2)*y(n-1) - ... - a(na+1)*y(n-na) If a(1) is not equal to 1, filter normalizes the filter coefficients by a(1).

C. Feature extraction

The features are derived from Linear Predictive Coding (LPC). LPC is a technique that attempts to derive the coefficients of a filter that would produce the utterance that is being studied. LPC is useful in speech processing because it has the ability to extract and store time varying formant information. Here formants are points in a sound's spectrum where the loudness is amplified. What we get from LPC analysis? We get a set of coefficients that describe a digital filter and this filter in conjunction with a noise source or a periodic signal that would produce a sound similar to the original speech. LPC data is often further processed to produce LPC-cepstrum features. The LPC-cepstrum vectors tend to place words that “sound” alike close together.

D. LPC: Linear Predictor Coefficients.

A = lpc(X,N) finds the coefficients, A=[1 A(2) ... A(N+1)], of an Nth order forward linear predictor. Xp(n) = -A(2)*X(n-1) - A(3)*X(n-2) - ... - A(N+1)*X(n-N) such that the sum of the squares of the errors err(n) = X(n) - Xp(n) is minimized. X can be a vector or a matrix which are the coefficients of the input speech and N specifies the order of the polynomial A(z) which is a positive integer. N must be less or equal to the length of X. If value for N is not specified then lpc uses a default N = length(X)-1. [A,E] = lpc(X,N) returns the variance or power of the prediction error. LPC uses the Levinson-Durbin recursion to solve the normal equations that is obtained from the least-squares formulation and this computation of the linear prediction coefficients is often referred to as the autocorrelation method.

E. Classification

A library of feature vectors is provided during “training” process. The classifier uses the feature vector of the input which is unknown word and attempts to find the “best” match from the library of known words which is trained during training process. The advanced recognizers uses classifiers that make use of HiddenMarkov Models (HMM) andArtificial Neural Networks (ANN).

IV. EXPERIMENTAL RESULTS

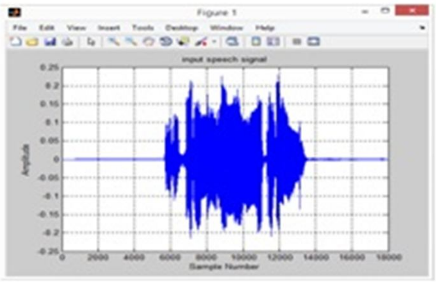

Figure 2.Input speech signal

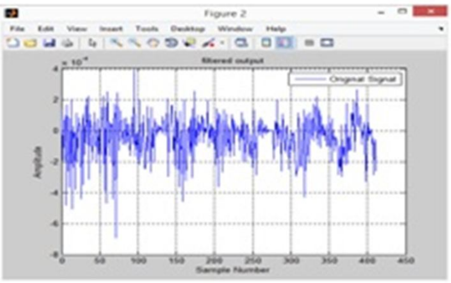

Figure 3.Filtered output

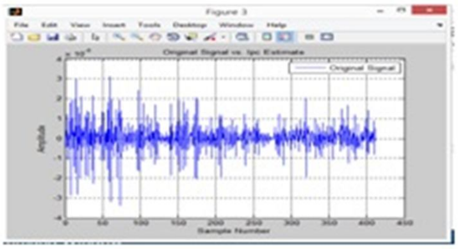

Figure 4.Original signal v/s lpc estimate

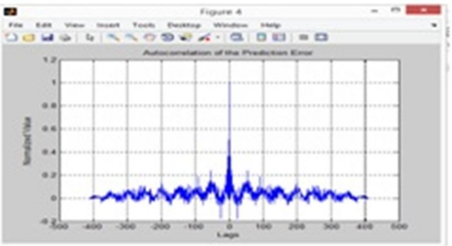

Figure 5.Autocorrelation of the prediction error

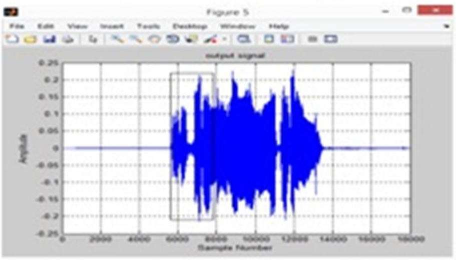

Figure 6.Predicted word “parisara”

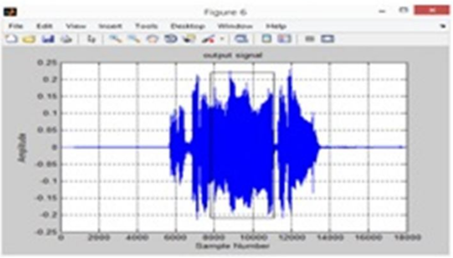

Figure 7.Predicted word “maalinyawanu”

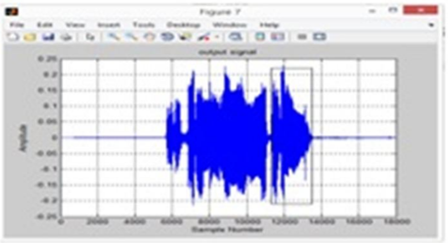

Figure 8.Predicted word “thadeyona”

V. DISCUSSION

The experimental results shown in the above figures is demonstrated on Kannada language. In Kannada language a sentence called "parisara maalinyawannu thadeyona" is recorded in .wav format. This file is then loaded into matlab and it is filtered out. Then only normalized value which lies between 0 and 1 is used for computation, this is because if the value is less than 0 then output result will be 0. Coefficients are then extracted from filtered output through LPC estimation. This LPC estimation will reduce the prediction error and then autocorrelation of the prediction error is computed. Finally based on coefficient matching of the word to be predicted is marked in input speech signal. The word prediction can be done automatically by using neural network method.

A. Advantages

- accelerate the writing
- reduce the effort needed to type
- Suggest the correct word (no misspellings).

B. Applications

- A combination of language model for word prediction can be useful for the physically disable persons and blind people.
- This word prediction using combination model can be applicable for any type of software and applications on combining a word n-gram and m-part of speech based language model.
- Voice recognition by computer is used in access control and security systems.
- Spelling Checkers
- Mobile Phone/PDA Texting
- Handwriting Recognition
- Word-sense Disambiguation

VI. CONCLUSION

This paper has proposed a method to improve the word prediction task based on LPC technique which combines a traditional word-based n-gram word model with a POS-based word model. The proposed methodology was evaluated on Kannada Language. The use of POS tags in language modelling helps to redefine the speech recognition problem so that we can find the best word sequence. This paper focuses on combining a word based n-gram and m-POS based word model to find the best matching word. It is also interesting to study the collision of the parameter on the proposed linear predictive model and present evaluations on different domain.

REFERENCES

[1] R. Foulds, *"Communication rates of non-speech expression as a functionin manual tasks and linguistic constraints."* in In Proceedings of the NInternational Conference on Rehabilitation Engineering. Toronto:RESNA, 1980, pp. 83–87.

[2] N. Garay-Vitoria and J. Abascal, *"Text prediction systems: a survey,"Univers. Access Inf. Soc.,* vol. 4, no. 3, pp. 188–203, Feb. 2006.

[3] M. Ghayoomi and S. Momtazi, *"An overview on the existing languagemodels for prediction systems as writing assistant tools,"* in Systems,Man and Cybernetics, 2009. SMC 2009. IEEE International Conferenceon, San Antonio, Texas, 11-14 October 2009, pp. 5083–5087, iSSN:

[4] 1062-922X.

[5] P. Vyrynen, *"Perspectives on the utility of linguistic knowledge inenglish word prediction,"* Ph.D. dissertation, University of Oulu, Linnanmaa,November 19th 2005.

[6] H. AlMubaid, *"A learning-classification based approach for word prediction,"*Int. Arab J. Inf. Technol., vol. 4, no. 3, pp. 264–271, 2007.

[7] N. Garay-Vitoria and J. Abascal, *"Modelling text prediction systemsin low- and high-inflected languages,"* Comput. Speech Lang., vol. 24,no. 2, pp. 117–135, 2010.

[8] J. L. Arnott and N. Alm, *"Towards the improvement of augmentativeand alternative communication through the modelling of conversation,"*Computer Speech and Language, vol. 27, no. 6, pp. 1194–1211, 2013,special Issue on Speech and Language.

[9] Bear, J. Dowding, and E. Shriberg. 1992. *Integrating multiple knowledge sources for detection and correction of repairs in human-computer dialog.* In Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics.

[10] Black, F. Jelinek, J. Lafferty, D. Magerman, R. Mercer, and S. Roukos. 1992*. towards history-based grammars: Using richer models for probabilistic parsing*. In Proceedings of the DARPASpeech and Natural Language Workshop.

[11] Breiman, J. Friedman, R.ichard A. Olshen, and C.harles J. Stone. 1984. *Classification andRegression Trees. Wadsworth & Brooks. P. Brown, V. Della Pietra, P. deSouza, J. Lai, and Robert L. Mercer. 1992*. Class-based n-gram models of natural language. Computational Linguistics.