Adaptive Hybrid POS Cache based Semantic Language Model

Manzoor Ahmad Chachoo
Department of Computer Sciences,
University of Kashmir
Srinagar, Kashmir -190006, India

S. M. K. Quadri
Department of Computer Sciences,
University of Kashmir
Srinagar , Kashmir -190006, India

ABSTRACT
This paper presents a language model as an improvement over the stochastic language model for developing a syntactic structure based on word dependencies in local and non local domain. The model copes with the issues of limited amount of training material and the exploitation of the linguistic constraints of the language. The proposed model is a dynamic probabilistic model which uses word dependencies based on their part of speech tags along with the tri-gram Model but also takes care of the influence of the word which are very far from the word being considered in a text and stores the word history in a dynamic cache for information mining using long distance dependency. The model based on second order Hidden Markov Model has been used and an improvement of 2% has been observed in the word error rate and 4% reduction in the perplexity when compared to the normal tri-gram model.

General Terms
Natural Language Processesing , Spoken Dialogue Systems , Computational Linguistics.

Keywords
Language Model , Dynamic language model , Part of Speech , POS Language Model , Word Dependencies , Speech recognition system.

1. INTRODUCTION

Language models captures the properties of a language and helps to predict a next word in the word sequence given the probabilities of the predecessor words which are calculated based on some given training text. The language model forms a very critical component for any spoken dialogue system as it defines the coverage and accuracy with which the system can understand what the user speaks and thus improving the performance of the dialogue manager. Statistical Language models also known as n-gram Language models characterize the word sequence as a Markov Process [1] meaning the probability of a word given all previous words depends on the immediately preceding words . A n-gram is a sequence of n symbols (e.g. words, syntactic categories etc) for some $n \geq 1$. When n= 2 it is known as bi-gram language model i.e. in a word sequence $w_1, w_2, \dots, w_n$ the word w_i is conditionally independent of the word history $w_1, w_2, \dots, w_{i-2}$ given the preceding word w_{i-1} .

$$P(w_i|w_{i-1}, w_{i-2}, \dots, w_1) = P(w_i|w_{i-1}) \dots . (1)$$

In this case the probability of the word sequence $P(w_1, w_2, \dots w_n)$ can be decomposed as the product of the conditional probabilities.

$$P(w_1, w_2, \dots w_n) = \prod_{i=1}^{n} P(w_i|w_{i-1}) \dots . \quad (2)$$

Estimates of Probabilities in n-gram models are commonly based on maximum likelihood estimates i.e. by counting the words in the document on some given training text. The conditional context component also referred to as history can be extended to consider more than one word e.g trigram language model which is given by the following equation

$$P(w_1, w_2, \dots w_n) = \prod_{i=1}^{n} P(w_i|w_{i-1}, w_{i-2}) \dots . (3)$$

The number of parameters in Markov Model is $|V|^n$ where V is the set of words , |V| size of the vocabulary and the order of the Markov process is n-1. The Markov parameters are typically estimated using in-domain text and the problem of storage space and attaching a reasonable degree of confidence to the derived estimates are to be considered. In most of the research domains a vocabulary size of 65000 words and n=3 also referred as trigram language models have given successful results but the related used words outside this two word context are not taken into consideration which can lead to improvement in the perplexity of the model.

2. CLASS BASED SEMANTIC MODEL

Due to the sparse training text , we make use of Equivalence class based n-gram model where the probability of word is dependent on its history via the words semantic class[2], groups of words that share a semantic category relevant to the spoken dialogue task . Considering some words as equivalent helps to reduce the word history equivalence classes to be modeled in the n-gram model. This is implemented by mapping a set of words to a word class by using a classification function. The domain knowledge can also be incorporated by classifying the relevant words into classes which may have some common feature e.g. In a medical assistance system, the user may select from a number of diseases which may be diagnosed based on a set of symptom. In this case we first select a set of semantic classes <diseases> , <symptoms> etc containing all the relevant diseases names and relevant symptoms appropriate for the domain and we then annotate the language model training corpus with the semantic classes: the training corpus is parsed with our natural language understanding grammar we find the constituents corresponding to the chosen semantic classes [3] . And then compute the probability distributions e.g $P(w| < diseases >)$ over all the words in the class. Consider a word sequence W = $w_1, w_2, \dots, w_n$ and let $C(w_i)$ be the class to which a word W_i belongs. The probability

$P(W)$ will be unique if the class are non overlapping else the probability of the word sequence using a trigram semantic class model is given by [4].

$$P(w_1, w_2, \ldots w_n) = \prod_{i=1}^{n} P\left(w_i \middle| C(w_i) P(C(w_i) | C(w_{i-2}) C(w_{i-1})\right) \ldots. \quad (4)$$

where $P\left(w_i \middle| C(w_i)\right)$ is the probability of the word w_i occurring in the semantic class $C(w_i)$. The probability distribution $P\left(w \middle| C(w)\right)$ depends on the semantic class. For instance, for the 'month' class we use the uniform distribution, but for the 'disease' class it is a function of the number of cases reported in the hospital as shown in Fig 1.

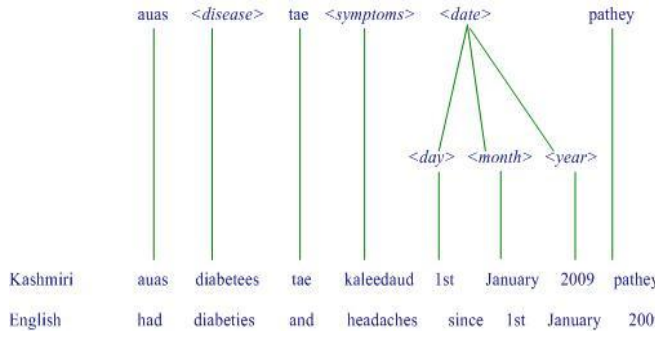

Fig 1: Semantic parse example

3. MODEL DESCRIPTION

Part-of-speech tagging is the act of assigning each word in a sentence a tag that describes how that word is used in the sentence. Typically, these tags indicate syntactic categories, such as noun or verb, and occasionally include additional feature information, such as number (singular or plural) and verb tense. Part of Speech Language Models have been used in speech recognition systems earlier [5][6][7] where the parameters are calculated using annotated training corpus.

Cache language model uses a window of the 'n' most recent words to determine the probability distribution of the next word [8]. To achieve the dynamic behavior the recent history has been stored and statistically evaluated in the caches earlier also[9][10]. In [9] the dynamic component used the a predicted POS in a trigram language model to adjust the probability of the next word. Each POS has a separate cache where the frequencies of all the word that occurred with a POS is used for the evaluation of the conditional probability of the next word. . As a word is observed it is tagged and the appropriate POS cache is updated.

The POS based Cache Semantic Model helps to identify the local dependencies between the words in a sequence based on the part of speech (POS) categories. The Parameters of POS model are of the form

$$P\left(w_i \middle| S(w_i)\right) \times P\left(S(w_i) | S(w_{i-2}) S(w_{i-1})\right) \ldots (5)$$

which means that the POS category $S(w_i)$ is first determined for a word w_i at position 'i' is based on the POS category $S(w_{i-2}) S(w_{i-1})$ of the two words that precede it. First the various POS Categories are defined in the form of a vector which can be enhanced later. Then the model is to be trained for which a large training text corpus is required along with each words all possible POS categories that the word can take. Various words of the suitably sized training text are annotated with the unambiguous part of speech (POS) categories since many words can have multiple POS categories depending upon their role in the text. Estimates of the frequency of the words in the vocabulary for setting the initial probability in the model.[11]. Hidden Markov models (HMM) are stochastic models capable of statistical learning and classification. They have been applied in speech recognition and handwriting recognition because of their great adaptability and versatility in handling sequential signals [12]. So we use second order Hidden Markov models where the states correspond to POS categories and are labeled by the category they represent. The A matrix contains state transition probabilities, the B matrix contains output symbol distributions, and the C matrix contains unknown word distributions. The probability of transitioning to a new state depends not only on the current state, but also on the previous state. This allows a more realistic context-dependence for the word tags than the first-order model. The elements of the output matrix have been assigned to word equivalence classes rather than the individual words which aid the estimation of the required number of parameters which is very large especially in different word types. Within these classes words have an uneven distribution and the transition matrix is set so that all the state transitions have an equal probability. The output matrix probability is based on the word occurrence probability $P(V_i)$ which is then converted to probabilities of the word equivalence classes $P(W_k)$. The probability of each equivalence class W_k is then divided equally among the POS categories that are in the equivalence class, to give weights $F(W_k, C_i)$. This reflects the assumption that all words in an equivalence class can initially function equiv-probably as any POS category of the class. The output matrix elements for each state are constructed using the various $F(W_k, C_i)$. For each state, the elements are then normalized to sum to unity.

The HMM model is then trained using Baum-Welch algorithm [13]. The algorithm (BW) is used for estimating the parameter values that maximize the likelihood of the training text belong to a family of algorithms called Expectation Maximization (EM) algorithms. They all work by guessing initial parameter values, then estimating the likelihood of the data under the current parameters. These likelihoods can then be used to re-estimate the parameters, iteratively until a local maximum is reached. To determine the most likely state sequence Viterbi algorithm [14] has been used which maximizes the probability of seeing the test sentence.

The static language model has a probability distribution for the next word conditioned on the previous words which are obtained by taking mean over many documents. The static model has a problem that some words or word sequence are more likely to happen within a specific context cannot depend on average over other documents . So to overcome this , we make use of a "dynamic" model based on a word cache which contains frequency ordered linked list of words occurring in the previous text history. In a specific topic various words tend to be repeated as such there frequency count is incremented or if the word is not in the list, the list is updated with a initial count of 1. These counts are used to determine the conditional probabilities of words in the dynamic cache which participates in determining the correlation with the previous two words.

4. MODEL ADAPTATION

In most natural language systems, the language used depends on the dynamic state / domain of the sentence. And for each state we collect a sub corpus that along with the semantic

class can guide a specific answer [2] and help the language model easily adapt to the changing needs of its applications. A well known method of adaptation is to build separate language models on the general and specific training data and then combine through linear interpolation [15]. If LG is the language model trained on the general training corpus and Ls is the language model trained on corpus specific to a state of the sentence. The likelihood of a word w_i given the two preceeding words w_{i-2}, w_{i-1} is estimated by the combined trigram language model as

$$P(w_i|w_{i-2}, w_{i-1}) = \lambda P_{LG}(w_i|w_{i-2}, w_{i-1}) + (1-\lambda)P_{LS}(w_i|w_{i-2}, w_{i-1}) \dots (6)$$

Where λ represents the interpolation weight which is trained on the developmental corpus to optimize word error rate and perplexity.

5. RESULTS AND DISCUSSION

We tested our model on a collection of test data sets using word error rate and perplexity reduction as our measure. Academic speech for advising(The MICASE corpus) from University of Michigan and The Trains corpus from University of Rochester were downloaded and used for the study .Our initial experiments focused on training the trigram language model on general corpusand then creating the semantic equivalence classes for the different words in the trigram model. Word error rates for two sentences that covered many training sentences is shown below.

	1	**2**	**Avg(1,2)**
Sentences	279	**293**	**572**
Words	926	847	1773
Tri-gram LM	24.5	32.4	28.45
Class Tri-gram LM	19.9	24.3	22.1
Adapative Class LM	18.5	21.7	20.1

. An improvement of 6.35% in the word error rate was observed when a POS cache based semantic class language model was used than a normal trigram model and a 2% improvement was observed when a adaptive trigram model was used. Also a 4% reduction in the perplexity was observed when compared to the normal tri-gram model.

6. CONCLUSIONS

We presented an adaptive hybrid model to improve over the stochastic language model. Our hybrid model uses the part of speech for the equivalence classes and aims at predicting the next word taking into account the semantic properties and the word dependencies by making use of the dynamic cache. The models gives lower perplexity on various tasks. We discussed the issues in constructing such a model and reported improvements in perplexity and word error rater(WER). The results reported are preliminary , we believe that the performance of the model can improve provided more fine tuning in the data structures based on lexical parsing is taken into account on which the work is underway. The conclusions we made in this paper are that the adapative cache based class semantic model is a powerful and flexible framework for language modeling attributed to using the class property in addition to the k- pervious word window.

7. ACKNOWLEDGMENTS

We thank the University of Michigan and University of Rochester for keeping the data sets online and free for academic and research usage.

8. REFERENCES

[1] L.R. Bahl, F. Jelinek and R.L.Mercer. 1983 . A Maximum Likelihood Approach to Continuous Speech Recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 5, pages 179-190,

[2] Roger Argiles Solsona, Eric Fosler-Lussier, Hong-Kwang J. Kuo, Alexandros Potamianos, Imed Zitouni, 2002, Adaptive language models for spoken dialogue systems" IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

[3] E. Fosler-Lussier and H.-K. J. Kuo,2001, Using semantic class information for rapid development of language models within ASR dialogue systems," in Proc. ICASSP, Salt Lake City, Utah.

[4] P.F. Brown, V.J. Della Pietra, P.V. DeSouza, J.C. Lai, and R.L. Mercer, 1992 Class-based n-gram models of natural language, Computational Linguistics, vol. 18, no. 4, pp. 467–479.

[5] P. Dumouchel, V.Gupta, M. Lennig & P. Mermelstein. 1988 , Three Probablisitic Language Models for a Large Vocabulary Speech Recognizer, Proceedings of International Conference on Acoustics Speech and Signal Processesing.

[6] K. Shikano, 1987, Improvement of Word Recognition Results by Trigram Model Proceedings of International Conference on Acoustics Speech and Signal Processesing.

[7] A.M. Derouault, B. Merialdo. 1986 , Natural Language Modeling for Phoneme to Text Transcription , IEEE Transactions on Pattern Analysis and Machine Intelligence Vol PAMI-8 No. 6.

[8] F. Jelinek, B. Merialdo, S. Roukos, and M. Strauss, 1991 , A Dynamic Language Model for Speech Recognition" In the Proceedings of the workshop on Speech and Natural Language, Association for Computational Linguistics Stroudsburg, PA, USA .

[9] Kuhn, R., 1988 ,Speech Recognition and the Frequency of Recently Used Words: a Modified Markov Model for Natural Language, Proceedings of COLING Budapest, Vol. 1, pp. 348-350.

[10] Kupiec, Julian , 1989 ,Probabilistic Models of Short and Long Distance Word Dependencies in Running Text', Proceedings of Speech and Natural Language DARPA Workshop, pp. 290-295.

[11] L.R. Rabiner, S.E. Levinson, and M.M. Sondhi, 1983. An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition. Bell System Technical Journal, Vol. 62, No.4, April. pp 1035-1074.

[12] Scott M. Thede , Mary P. Harper , 1999 , A second-order Hidden Markov Model for part-of-speech tagging',In the Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics..

[13] L.E. Baum., 1972 ,An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of a Markov Process. Inequalities, 3,. pp. 1-8.

[14] A. J. Viterbi. , 1967 , Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm. IEEE Trans. on Information Theory Vol. IT-13, April. pp. 260-269.

[15] F. Jelinek. 1991 , Up from trigrams ! the struggle for improved language models, in Proc EUROSPEECH pp 1937-1040