Volume 3, No. 6, June 2012
Journal of Global Research in Computer Science
ISSN-2229-371X
RESEARCH PAPER
Available Online at www.jgrcs.info

SEMI-SUPERVISED LEARNING OF UTTERANCES USING HIDDEN VECTOR STATE LANGUAGE MODEL

Manzoor Ahmad Chachoo*[1] Dr. S. M. K. Quadri[2]
Department of Computer Sciences , University of Kashmir Srinagar , Kashmir -190006 , India
manzoor@kashmiruniversity.ac.in*[1]
Department of Computer Sciences , University of Kashmir Srinagar , Kashmir -190006 , India
Quadrismk@hotmail.com

Abstract: Spoken dialogue system has an uncertain parameter during the speech recognition which controls its performance that vary for the different users as well as for the same user during multiple repetitions of even the same dialogue. This paper discusses how recognition errors in the users utterances can be handled by making use of semi-supervised learning techniques over the hidden vector state (HVS) model. The HVS Model is an extension of basic Markov model in which the context is encoded in each state as a vector. The state transitions in the HVS are factored into a stack shift operation similar to the push-down automaton. HVS-Model being a statistical model requires lot of labeled training data which is practically difficult. In this paper we present how classification and expectation-maximization semi-supervised learning approaches can be trained on both labeled and unlabelled corpora for handling the uncertainty by the user as well as the recognition errors by speech recognition system. The experimental results show that the proposed framework using the HVS model can improve the performance of the dialogue management of the spoken dialogue system when compared with the baseline model.

Keywords: Spoken dialogue system , Speech recognition system . Machine Learning, Expectation Maximization , classification . weighted minimum edit distance.

INTRODUCTION

Spoken Language Understand has been a challenge in the design of the spoken dialogue system where the intention of the speaker has to be identified from the words used in his utterances. Typically a spoken dialogue system comprises a four main components an automatic speech recognition system (ASR) , Spoken language understanding component (SLU) , Dialogue manager (DM) and an Speech synthesis system which converts the text to speech (TTS). Spoken Language understanding deals with understanding the intent from the words of the speakers utterances. The accuracy of the speech recognition system is questionable and researchers have provided various solutions to the problem and classifying the information may actually guide the dialogue manager in framing a response.

Many models both statistical as well as empirical methods have been suggested for extracting information from text by automatically generating a language model after training from the annotated corpus.[1] When Statistical classifiers are used for classification they have to be trained using a large amount of task data which is usually transcribed and then assigned one or more predefined type to each utterance by humans , a very expensive and laborious process[2]. But they do not perform well due to the lack of large scale richly annotated corpora. Seymore et al [3] extracted the important information from the headers of computer science research papers by making use of Hidden Markov models. A statistical method based on HVS has been proposed to automatically extract information related to protein – protein interactions from biomedical literature [2].

Semi-supervised learning uses both supervised and unsupervised learning to learn from both annotated and unannotated sentences for classifications , clustering and so on. Nigam et al [4] used Expectation-Maximization algorithm with a naïve Bayes classifier on multiple mixture components for text classification. Small amount of labeled data is used to first build a model which is then used to annotate the instances of the unlabeled instances. The instance along with identified label which posses the more confidence measure are then added to the training set and participate in retraining of the model for the left out instances. The process is continued for the training of the remaining of the un-annotated sentences.

THE HIDDEN VECTOR STATE MODEL

The basic hidden vector state model is a discrete Hidden Markov Model in which each HMM state represents the state of a push down automaton which encodes history in a fixed dimension stack . Each state consists of a stack where each element of the stack is a label chosen from a finite set of cardinality $M+1$ $C=\{c_1,..,c_M,c_\#\}$. A HVS model state of depth D can be characterized by a vector of dimension D with most recently pushed element at index 1 and the oldest at index D. Each vector state is like a snapshot of the stack in the push-down automaton and transitions between states can be factored into a stack shift by 'n' positions followed by a push of one or more new pre-terminal semantic concepts. The number of new concepts to be pushed is limited to one. The joint probability $P(W,C,N|\lambda)$ of a sequence of stack pop operations , word sequence W and concept vector sequence C is approximated as

$$P(W,C,N) = \prod_{t=1}^{T} P(n_t|W_1^{t-1},C_1^{t-1}) \,.\, P(C_t[1]|W_1^{t-1},n_t) \,.\, P(C_t[1]|W_1^{t-1},n_t) \ldots (1)$$

with the assumptions as

$$P(n_t|W_1^{t-1},C_1^{t-1}) \approx P(n_t|c_{t-1})$$

$$P(C_t[1]|W_1^{t-1}, n_t) \approx P(c_t[1]|\ c_t[2 \ldots D_t])$$
$$P(C_t[1]|W_1^{t-1}, n_t) \approx P(w_i|\ c_i)$$

so we have

$$P(W, C, N) = \prod_{t=1}^{T} P(n_t|c_{t-1})\ .\ P(c_t[1]|\ c_t[2 \ldots D_t])\ .\ P(w_i|\ c_i) \ldots .(2)$$

Where

a) c_t denotes the vector state at word position t , which consists of D_t semantic concept labels (tags) , i.e. $c_t = [c_t[1], c_t[2] \ldots, c_t[D_t])$ where $c_t[1]$ is the preterminal root and $c_t[D_t]$ is the root concept normally represented by SS (Sentence Start).

b) n_t is the vector stack shift operation and takes values in the range of $0 \ldots D_{t-1}$ where D_{t-1} is the stack size at word position $t-1$.

c) $c_t[1] = c_{w_t}$ is the new preterminal semantic tag assigned to word w_t at word position t.

The key feature of the HVS model is its ability for representing hierarchical information in a constrained way which can be trained from only lightly annotated data. The generative process associated with HVS model consists of three steps for each position t :

a) Choose a value for n_t .

b) Select preterminal concept tag $c_t[1]$.

c) Select a word w_t.

A set of domain specific lexical classes and abstract semantic annotations which limit the forward and backward search to include only those states which are consistent with these constraints for the model training must be provided for each sentence.

SEMI-SUPERVISED LEARNING

The main aim of the semi-supervised learning is to utilize the labeled utterances for annotating the unlabelled utterances in order to improve the performance of a classifier and reducing the human labeling effort. The semi-supervised learning technique used is as follows , Initially the human labeled task data is used to train the initial model which is used then to classify the unlabelled utterances. The machine labeled utterances whose confidence score value is above a threshold so that the noise due to classifier errors is reduced are added to the training data .If the input space is X and the output is $\{-1, 1\}$ it is known as binary classification. Suppose E_L is the small set of labeled sentences $\{< s_1, a_1 >, < s_2, a_2 >, \ldots, < s_i, a_i >$ where $S = \{s_1, s_2, \ldots, s_i\}$ is the set of sentences and $A = \{a_1, a_2, \ldots, a_i\}$ is the set of corresponding annotation for each sentence . And E_u is the large set of unlabelled data $E_u = \{\ s_{i+1}\ ,\ s_{i+2}\ , \ldots \ldots s_{i+u}\}$. The process of predicting the labels A_u of the unlabelled data S_u is known as the transduction. The process of constructing a classifier $f : X = \{-1, 1\}$ on the whole input space using the unlabeled data comes under the purview of semi-supervised learning.

RELATED WORK

In Language Processesing framework there are two approaches viz certainty based approaches and committee based approaches of having control over the type of inputs on which it trains [6] .In certainty based approaches , a small set of annotated examples is used to train the system, the system then labels the unannotated sentences and then determines the confidence for each of its prediction. The sentences with lower confidence are then presented to the labelers for annotation. In Committee based methods , a small set of annotated sentences are used to create a disjoint set of classifiers, which are then used to classify the unannotated sentences. The sentences where the classification differ much are manually annotated . Nigam et al (2000) learned from both labeled and unlabelled data based on combination of Expectation Maximization and a Naïve Bayes classifier on multiple mixture components per class for task of text classification. Yarosky [6] used self training for word sense disambiguation. Rosenberg et al [7] applied self training to object detection from images. Self training builds a model based on the small amount of labeled data and then uses the model to label instances in the unlabeled data . The most confident instances together with their labels participate in the training set to retrain the model.

Ghani(2002) proposed an algorithm for exploiting the labeled as well as un- labeled data using the co training with Expectation Maximization(CO-EM)[8]. Riccardi and Hakkani –Tur(2003) used semi-supervised learning for automation speech recognition and have shown improvements for statistical language modeling where they exploited confidence scores for words and utterances computed from ASR word lattices[9].

FRAMEWORK

A probabilistic framework is used to describe the nature of sentences and their annotations where semantic annotations are considered as the class label $g \in G$ for each sentence with the following two assumptions a) If $|G|$ is the number of distinct annotations in the labeled set E_L where $E_L = \{$ $(\{\ (s_1, a_1), (s_2, a_2), \ldots .(s_L, a_L)\}$ then the data are produced by $|G|$ probability models. b) there is a one to one correspondence between probability components and classes. Considering the each individual annotation as a class , the likelihood of a sentence s_i is given by

$$P(s_i|\lambda) = P(\ a_i =\ g_j|\lambda) P(s_i|a_i =\ g_{j\ ,}\ \lambda)$$

Where g_j is the annotation of the sentence s_i and λ represents the complete set of HVS model parameters. Since the domain of possible training examples is $s_{|L|+|U|}$ and the binary indicators are known for the sentences in E_L and unknown for the sentences in E_U. The class labels of the sentences are represented as the matrix of binary indicators Z where

$$z_{ij} = \begin{cases} +1 & if\ \ a_i =\ g_{j\ ,} \\ 0 & other\ wise \end{cases}$$

Then we have

$$P(s_i|\lambda) = \sum_{j=1}^{|G|} z_{ij}\ P(\ g_j|\lambda) P(s_i|\ g_{j\ ,}\ \lambda)$$

Calculating the maximum likelihood estimate of the parameters λ i.e. $argmax_\lambda\ P(W, C, N \mid \lambda)$ for learning the HVS model . The annotation A for the word sequence W

can be determined by $\{C, N\}$ i.e the concept vector sequence C and the series of stack shift operations N and $\{C, N\}$ can be inferred from A. Thus $argmax_\lambda P(W, C, N \mid \lambda)$ can be rewritten as $argmax_\lambda P(W, A \mid \lambda)$ which can further be rewritten as $argmax_\lambda P(E \mid \lambda)$ which is the product over all the sentences assuming each sentence is independent of each other . The probability of the data is given by

$$P(E|\lambda, Z) = \prod_{s_i \in E} \sum_{j=1}^{|G|} z_{ij} P(\ g_j|\lambda) P(s_i|\ g_{j,}\lambda)$$

The complete log likelihood of the parameters $l_g\ (E|\lambda, Z)$ can be expressed as

$$l_g(E|\lambda, Z) = \sum_{s_i \in E} \sum_{j=1}^{|G|} z_{ij}\, log\ P(\ g_j|\lambda) P(s_i|\ g_{j,}\lambda)$$

METHODOLOGY

To improve the performance of classifier, the methods used are based on classification and Expectation Maximization. Both the methods assume that there is some training data available for the initial classifier. The main aim is to use this classifier to label the unlabelled data automatically and to then improve the classifier performance using machine labeled utterances. Semi-supervised learning based on classification measures the edit distance between the POS tag sequences of the sentences in E_L and POS tag sequences of sentences in E_U to automatically generate the annotation for the unlabelled sentences. The edit distance or *Levenshtein distance* of two strings, s1 and s2, is defined as the minimum number of point mutations required to change s1 into s2, where a point mutation can be either changing a letter or inserting a letter or deleting a letter .If *X* and *Y* are two pos tag sequences of length *n* and *m* respectively , a tabular computation *D(i , j)* which contains the score of the optimal alignment between the initial segment from *X* and the initial segment from *Y* is calculated using the following algorithm.

a. Edit_Distance(X,Y)
b. Initialize
   a) $\mathrm{D(i, 0) = \ i}$ and
   b) $\mathrm{D(0, j) = \ j}$
c. Recurrence relation
   a) for each $\mathrm{i = 1 \dots m}$
   b) for each $\mathrm{j = 1 \dots n}$

$$\mathrm{D(i,j)} = \begin{cases} \mathrm{D(i-1\,,j) + \ 1} \\ \mathrm{D(i\,,j-1\,) + \ 1} \\ \mathrm{D(i-1\,,j-1) +} \begin{cases} 0 & \text{if X(i) = Y(j)} \\ 2 & \text{if X(i)} \neq \text{Y(j)} \end{cases} \end{cases}$$

d. Termination
   $\mathrm{D(n, m)}$ is the minimum edit distance

Dynamic programming which solves problems by combining solutions to sub problems is used comprising of edit distance matrix $D(i,j)$. By this technique we first calculate $D(i,j)$ for smaller *i , j* and compute larger *D(i , j)* based on the previous computed smaller values i.e compute $D(i,j)$ for all $0 < i < n$ and $0 < j < m$. Given two sentences S_i, S_j and their corresponding POS tag sequences $T_i = a_1a_2 .. a_{ni}$ and $T_j = b_1b_2..b_{nj}$, the distance between the two sentences is defined as $Dist(S_i, S_j) = - D(n_i, n_j)$ where $D(n_i, n_j)$ is the distance measure of optimal alignment between two POS tag sequences T_i and T_j.

DISTANCE-WEIGHTED NEAREST NEIGHBOR ALGORITHM

Classification a spoken dialogue learning uses a finite number of labeled examples and selects a hypothesis is expected to generate few errors on the future examples. In case of spoken dialogue systems human labeling of the the spoken utterances has a wide impact on the quality of the machine labeling of the unlabeled sentences . The basic elements to handle by classification algorithm are word lattices which may contain a single word or a collection of words with some weight or probability [10]. The technique which we have used for classification is Distance-Weighted Nearest Neighbor Algorithm. Since the training Input variables consists of the set <X ,Y> where X contains represents the word and Y represents its semantic annotation, the algorithm find s the training points which have the closest edit distance to the queried word. It assigns weights to the neighbors based on their 'distance' from the query point, the Weight are inverse square of the distances. and then classifies according to the mean value of the 'k' nearest training examples. All the training points influence a particular instance.

Transductive Learning based on expectation maximization:

The EM algorithm is an efficient iterative procedure to compute the Maximum likelihood (ML) estimate in the presence of missing or hidden data. In ML estimation, we wish to estimate the model parameter(s) for which the observed data are the most likely. So we cluster the sentences in E_l and E_u. The original model will contain more sentences since some sentences in E_u will have the similar semantic structure with those sentences in E_l which have been used to train the HVS Model but adding should be based on some confidence measure so that the performance of the model is improved. To do this a parameter DG_f which represents the degree of fitness is to be used for selecting the sentences DG_f based on parsing information I_p, structural information I_s and complexity information I_c[2]. These parameters of a sentence are defined as

Parsing information I_p describes the information in the parsing result and is defined as

$$I_p = 1 - \frac{\sum_{j=1}^{N} KEYI(s_{ij})}{\sum_{j=1}^{N} KEY(s_{ij})}$$

Where N denotes the length of the sentence s_i , s_{ij} denotes the j[th] word of the sentence s_i and the functions $KEYI(s_{ij})$ is equal to 1 if s_{ij} is a word in the E_l and 0 otherwise. $KEYI(s_{ij})$ is 1 if $KEYI(s_{ij})$ is 1 and the semantic tag of (s_{ij}) is not known and 0 otherwise.

Structure information I_s is a measure of similarity between the structure information of a sentence s_i and the sentences s_j in E_l which is given by

$$I_s = 1 - \frac{min(Dist(s_i, s_j))}{\max(Dist(s_k, s_j))} + \frac{NUM(C(s_i))}{|E_l|}$$

Where $s_j \in E_l$ and $s_k \in E_u$, $C(s_i)$ denotes the cluster where s_i is located, $(Dist(s_i, s_j))$ is the edit distance measure between sentence s_i and s_j. $NUM(C(s_i))$ is the number of sentences in the cluster $C(s_i)$.

Complexity information I_c is based on the length of the sentence s_i and the max length of the sentence s_j *where* $s_j \in E_l \cup E_u$. I_c is given by

$$I_c = 1 - \frac{length(s_i)}{\max(length(s_j) | s_j \in E_l \cup E_u)}$$

Since the measure of selecting a sentence is based on the degree of fitness DG_f which is given by

$$DG_f = \beta_p I_p + \beta_s I_s + \beta_c I_c + \beta_o$$

The coefficients $\beta = (\beta_p, \beta_s, \beta_c, \beta_o)$ are calculated using the method of least squares and β is selected to minimize the residual sum of squares.

$$RSS(\beta) = \sum_{i=1}^{N} (DG_f' - DG_f)^2$$

The parameter β is estimated from the N set of training data, DG_f' is the estimated value and DG_f is the observed value. First a sample corpus of words are identified from the travel domain. Then a semantic tag based on the class is attached for identifying interactions. The vertibi decoding algorithm is used to parse the sentences of the E_L. For the sentences in E_U selection is done based on the parameters i.e. DG_f. Thus the sentences in E_U would be added to the set of sentences with annotation and participate in further automatically annotating sentences in E_U.

EXPERIMENTS

To evaluate the models proposed the training data was split into two data sets corpus I comprising of 200 sentences out of which 100 sentences with manual annotation from travel domain are added to E_L for training the HVS model and 100 sentences were added to E_U. First clusters are created from the learned sentences based on the edit distance measure and then semi supervised learning based on expectation maximization was applied to the sentences in E_U The corpus II comprised of the 250 sentences which incremented the 200 sentences by 50 more sentences with annotation for learning the HVS model. And then out of 100 sentences 47 sentences were semantic annotated successfully with out any human labeling by the algorithm.

RESULTS

The experimental results for the baseline HVS model trained on sentences in E_L contained 74 classes when classification was performed . 8 Experiments were performed for subset of sentences in E_U with the k = 1,2,3 based on Distance-Weighted Nearest Neighbor Algorithm. The overall precision was calculated by ratio of (Number of sentences for which annotation was done correctly)/SUM (Number of sentences for which annotation was done correctly , Number of sentences for which annotation was done Incorrectly) based on classification. The overall precision in the travel domain data set was observed at 65.4% with k=3 when only sentences from E_L were used. The HVS Model was incrementally trained with these newly added sentences from E_U based on the sentence selection based on expectation maximization which improved the performance by 4.6%

Table: 1

Experiment	Precision %(E_L)	Precision % ($E_U + E_L$)
1	54.3	62.1
2	58.7	59.6
3	59.9	61.7
4	**64.1**	65.4
5	65.7	59.2
6	57.1	**68.7**
7	58.2	65.8
8	52.3	67.3

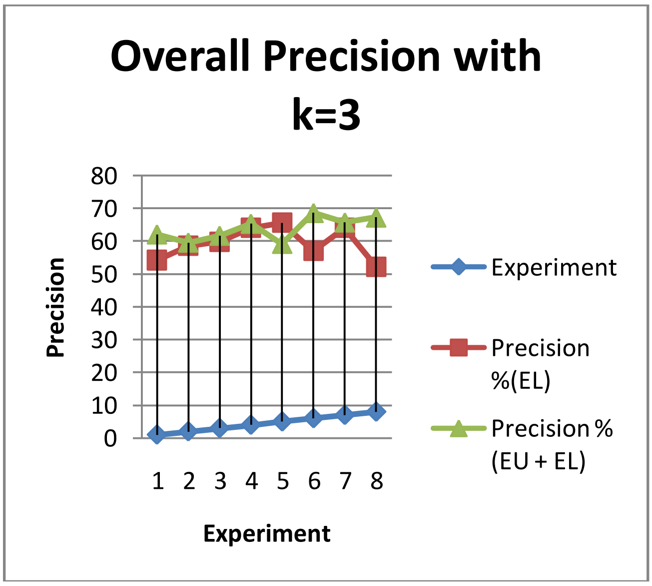

Figure: 1

CONCLUSION AND FUTURE WORK

In this paper we have used two semi-supervised learning techniques which made use of both labeled and unlabeled data to improve the performance of the HVS model. The overall performance was improved by nearly 4-5% . In future we will use the learning technique like SVM or Kernels for dealing with problems where minimum labeled data is available.

ACKNOWLEDGEMENT

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

REFERENCES

[1]. G. Tur , Dilek Hakkani- Tur , Robert E. Schapire , “Combining active and semi-supervised learning for spoken language understand, 175-186 Speech communications.

[2]. Deyu Zhou , Yulan He , Chee Keong Kwoh ,” Semi Supervised Learning of Hidden Vector State model for extracting protein-protein interactions, 209-222 , Artificial intelligence in Medicine (2007) vol – 41.

[3]. Seymore K, McCallum A, Rosenfeld R. Learning hidden Markov model structure for information extraction. In: Proceedings of the sixteenth national conference on artificial intelligence (AAAI-99)

[4]. Zhou D, He Y, Kwoh CK. Extracting protein—protein interactionsfrom the literature using the hidden vector state model. In: Alexandrov VN, van Albada GD, Sloot PMA, Dongarra J, editors. Lecture notes in computer science, vol. 3992. 2006. p. 549—56.

[5]. Nigam K, McCallum AK, Thrun S, Mitchell TM. Text classification from labeled and unlabeled documents using EM. Machine Learn 2000; 39(2/3):103—34.

[6]. Yarowsky D. Unsupervised word sense disambiguation rivaling supervised methods. In Proceedings of the 33rd annual meeting of the association for computational linguistics. Morristown, NJ, USA: Association for Computational Linguistics; 1995. p. 189—96.

[7]. Rosenberg C, Hebert M, Schneiderman H. Semi-supervised self-training of object detection models. In: Proceedings of the seventh IEEE workshop on applications of computer vision. Washington, DC, USA: IEEE Computer Society; 2005. p.29—36.

[8]. Ghani, R., July 2002. Combining labeled and unlabeled data for multiclass text categorization. In: Proc. Internat. Conf. on Machine Learning (ICML), Sydney, Australia.

[9]. Riccardi, G,]Gorin, A.L Wright, J.H., 2002. Automated natural spoken dialog. IEEE Computer. Mag. 35 (4), 51–56.

[10]. Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. In Proceedings of Annual workshop on computational learning theory,

[11]. Argamon-Engelson, S., Dagan, I., 1999. Committee-based sample selection for probabilistic classifiers. J. Artif. Intell.Res. 11, 335–360.

[12]. Blum, A., Mitchell, T., July 1998. Combining labeled and unlabeled data with co-training. In: Proc.of the eleventh annual conference on computational learning theory. New York, NY, USA: ACM Press; 1998 . p.92—100.

[13]. Jones R. Learning to extract entities from labeledand unlabeled text. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, 2005.

[14]. Blum A, Chawla S. Learning from labeled and unlabeled data using graph mincuts. In: Brodley CE, Danyluk AP, editors. Proceedings of the 18th international conference on machine learning. Morgan Kaufmann; 2001. p. 19—26.

[15]. Zhu X. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences Department, University of Wisconsin-Madison; 2005.

[16]. He Y, Young S. Semantic processing using the hidden vector state model. Comput Speech Lang 2005;19(1):85—106.

[17]. Cohn, D., Atlas, L., Ladner, R., 1994. Improving generalization with active learning. Machine Learning 15, 201–221.

[18]. Freund, Y., Seung, H.S., Shamir, E., Tishby, N., 1997. Selective sampling using the query by committee algorithm. Machine Learning 28, 133–168.

[19]. Iyer, R., Gish, H., McCarthy, D., May 2002. Unsupervised training techniques for natural language call routing. In: Proc. Internat. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Orlando, FL.