http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (8): 1190-1198, 2007
ISSN 1812-5638

Hidden Markov Model Based Part of Speech Tagger for Urdu

Waqas Anwar, Xuan Wang, LuLi and Xiaolong Wang
School of Computer Science and Technology,
Harbin Institute of Technology, Shenzhen Graduate School, China

Abstract: In this study, we present the preliminary achievement of Hidden Markov Model (HMM) to solve the part of speech tagging problem of Urdu language. The presented HMM is derived from the combination of lexical and transition probabilities. An important feature of our tagger is to combine many distinguished smoothing techniques with HMM model to resolve the data sparseness problem. We note that the proposed HMM based Urdu Part of speech tagger with different smoothing method has achieved significant performance. We evaluate our tagger's results regarding different smoothing methods and different word level accuracy through Analysis of Variance (ANOVA) and show how present results are significant. Also, we compose a confusion matrix about most frequent error occurring tag pairs. The development of our tagger is an important milestone toward Urdu language processing. This will open some novel research directions to mature Urdu language processing.

Key words: Urdu language, hidden Markov model, smoothing methods, part of speech tagging

INTRODUCTION

Parts Of Speech (POS) tagging is a process of assigning accurate syntactic categories (noun, verb, adjective etc.) to every word in the text (Jurafsky and Martin, 2000) and plays fundamental role in various Natural Language Processing (NLP) applications such as speech recognition, information extraction, machine translation and word sense disambiguation etc. POS tagging particularly plays very important role in word-free languages because such languages have relatively complex morphological structure of sentences than other languages. Indic and Urdu are good candidate examples of such word-free languages. Although POS-tagging for Indic languages has gained an increased interest over the past few years, yet the lack of availability of annotated corpora resources hinder the research and investigations, beside other disambiguation problems. Standardization is another problem because so far no standard tag sets are available for such languages. While so far this is the situation for Indic languages, Urdu has relatively more issues as it is quite far less studied and researched.

HMM is one of the distinguished probabilistic models used to work out a number of different problems and hence also repeatedly used in language processing problems. Specifically for the case of disambiguation issues, HMM has been effectively utilized to find out most probable state sequence for a particular sentence (Fatima and Guessoum, 2006). In this study, we also attempt to resolve the Urdu language processing disambiguation problem through HMM based model. The choice of HMM over other probabilistic language models has numerous motivations (Fatima and Guessoum, 2006). For example, HMM is not only well suited for modeling of sequential data, such as spoken or written language, further it also has strong statistical and theoretical background to construct models for text based tasks. (Sigletos *et al*., 2002). Here we try integrating different smoothing techniques with HMM to achieve significant results about disambiguation. Furthermore, we also apply Viterbi algorithm to assign the most likely POS-tag to every word in the corpus.

OVERVIEW OF URDU LANGUAGE

Urdu is a derivative word from Turkish mean horde (Lashkar). Urdu, an Indo-European language of the Indo Aryan family, is spoken in India and Pakistan. It is the national language of Pakistan having eleven million speakers. Among all languages in the world it has very close similarity to Hindi language. Urdu and Hindi both have originated from the dialect of Delhi region and

Corresponding Author: Waqas Anwar, School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen Graduate School, China

Table 1: Word order and semantic meaningfulness in Urdu language

Sentence	Correctness	Sentence	Correctness
بندر پیڑ کے اوپر بیٹھا ہے	T	Monkey tree on sitting	F
پیڑ کے اوپر بندر بیٹھا ہے	T	Tree on monkey sitting	F
اوپر پیڑ کے بندر بیٹھا ہے	T	On tree monkey sitting	F
بندر ہے بیٹھا اوپر پیڑ کے	T	Monkey is sitting on tree	T
پیڑ کے اوپر بندر ہے بیٹھا	T	Tree on monkey sitting	F
بیٹھا ہے بندر پیڑ کے اوپر	T	Sitting is monkey tree on	F

True = T, False = F

beside few minute details these languages share morphology. Since Hindi has adopted many words from Sanskrit, Urdu has also borrowed a large number of vocabulary items from Persian and Arabic. Urdu is also borrowing a number of vocabularies from Turkish, Portuguese and English. Furthermore several Arabic words have been borrowed by Urdu through Persian language. These words vary slightly in their tone, connotations and feelings. One of the noteworthy aspects of Urdu grammar constitution is its word order SOV (subject, object and verb) (Hardie, 2003)

Word order: Urdu is a word-free order language as compared to other languages, like English and European. Following example presents a clearly demonstration of free-word characteristic of Urdu (Javed, 1985) (Table 1).

Previous work: A variety of techniques have been used for automatic part of speech tagging and broadly these can be classified into two major categories: Statistical and Rule Based. Different techniques used for automatic part of speech tagging and applied to English and other European languages comprise of ruled based, Statistical based (Hidden Markov Model, Maximum Entropy and Conditional Random Field), finite-state transducers and neural network based approaches.

The classical techniques for designing POS consist of two stage architecture. The pioneering researcher like Harris, Kelin, Simmons, Greene, Rubin used the same architecture. The first phase of this system is to apply dictionary and to assign all probable part of speech tag to every word. The second phase employs a number of hand-crafted disambiguation rules to find out most appropriate tag for each word (Jurafsky and Martin, 2000).

Stochastic approaches to POS-tagging is not a new one since during 1980s most study of Marshall, Church, Derose, Merialdo and Brants have focused on stochastic based tagging. The other notable language models in Part of speech tagging are Brill transform based learning algorithm, Daelemans memory based tagging algorithm (Jurafsky and Martin, 2000).

The above mentioned taggers and tagging techniques have been used for English, European and some of East Asian languages. Although South Asian languages have big community all over the world but still most of language processing research has focused on other Asian languages. In this regard Urdu language processing is specifically quite far less studied and researched and therefore quite a limited work has been carried out on Urdu language processing.

To the author's best knowledge there is only one POS-tagger for Urdu developed by Hardie (2003). POS tagset used by this POS-tagger uses grammar of Urdu by Schmidt with EAGLE guideline morphosynatics annotation. It has uni-rule disambiguator having approximately 270 written rules.

Hidden Markov Model for part of speech tagging: HMM was first introduced by Rabiner (1989) while later Scott redefined it for POS tagging. Here below we outline the basic elements of HMM as defined for POS tagging problem by Scott and Harper (1999), for better understanding of HMM model we have used the same symbol as used by Scott and Harper (1999). The HMM is implemented in Bird *et al.* (2007).

N = The number of distinct states in the model. In terms of POS tagging N is the total number of tags that can be used by the tagger. Every possible state of HMM is represented by a tag in POS tagging problem.

M = The number of distinct output symbols in the alphabet of the HMM. While defining M for POS tagging problem we say that M is the total number of words in the lexicon of the tagger.

$A = \{a_{ij}\}$ = The state transition probability distribution. According to HMM this is the probability that a process will move from state i to another state j in one transition. In terms of defining this state transition probability distribution for POS tagging problem we say that it's the probability that the model will move from t_i to t_j.

$B = \{b_j(k)\}$ = B is the observation symbol probability (emission probability) distribution. This is the probability when the model is in a particular state j the k_{th} output symbol will be emitted. In term of POS tagging it's the probability that word w_k will be excluded when the tagger is at tag t_j (i.e., $p(w_k|t_j)$.

$\pi = (\pi_i\}$ = The initial state distribution. It is the probability that model will start in state i. In terms of POS tagging we say that this is the probability that the sentence will begin with tag t_i.

The presented model is a type of first order HMM, also referred to as bigram POS tagging. For POS-tagging problem presented Hidden Markov Model is composed of two probabilities: lexical (emission) probability and contextual (transition) probability (Samuelsson, 1996).

$$(t_1,...,t_n)^* = \underset{t_1,...t_n}{\text{argmax}}\ P(t_1,....,t_n)|(w_0,...,w_n)$$

Using Baye's law above equation can be rewritten as:

$$P(t_1,...,t_n|w_1,...,w_n) = P(t_1,...,t_n) \text{ x } \frac{P(w_1,...,w_n|t_1,...,t_n)}{P(w_1,...,w_n)}$$

$$(t_1,...,t_n)^* = \underset{t_1,...,t_n}{\text{argmax}}\ P(t_1,...,t_n) \text{ x } P(w_1,...,w_n|t_1,...,t_n)$$

$$(t_1,...t_n)^* = \underset{t_1,....t_n}{\text{argmax}}\ P(t_1,....t_n) \text{ x } P(w_1,...w_n|t_1,...t_n)$$

$$= \underset{t_1,....t_n}{\text{argmax}} \prod_{i=1}^{n} (\underbrace{P(t_i|t_{i-1})}_{\text{Transition probability}} * \underbrace{P(w_i|t_i)}_{\text{Emission probability}})$$

During training process our Urdu tagger will be trained through machine readable tagged corpora with partially morph syntactic tagset. The bigram language model computes two probability factors of the sequences during training process. While lexical probabilities are used to find the probability of a particular tag conditioned on particular word, contextual probabilities are aimed at determining the probability of a particular tag conditioned on immediate preceding tag (Samuelsson, 1996). After evaluating these probabilities respective distributions are stored in A and B (Fatima and Guessoum, 2006). Since we use Maximum Likelihood Estimator (MLE) as the baseline algorithm for estimating the values therefore during training process model may encounter problems for the situations where words or tag sequences do not occur or have a quite low frequency of occurrence. Consequently if the probability of occurrence of some unknown word or tag sequence is zero then entire observed sequence may assume a zero value, leading to data sparseness situation. In order to avoid data sparseness problem we introduce different smoothing techniques with HMM. Introduction of smoothing techniques not only yield significant improvement of the accuracy of unknown words but also the overall accuracy, as it will be explained in the following section.

Having computed the transition and emission probabilities and assigning all possible tag sequences to all the input words, now we are in the situation to construct a lattice showing association of different tags to the input words. Now we need an algorithm that can search the tagging sequence and maximize the product of transition and emission probabilities. For this purpose we use Viterbi algorithm, a dynamic programming process, which compute the maximized as well as optimal tag sequence with best score. The pseudo code of our Viterbi algorithm is shown as: (Manning and Schutze, 1999).

(1) Comment:Initialization
(2) $viterbi_1(PERIOD):=1.0$
(3) $viterbi_1(t):=0.0$ for $t \neq PERIOD$
(4) Comment:Induction
(5) for i:=1 to n step 1 do
(6) for all tags t_j do
(7) $viterbi_{i+1}(t_j):=max_{1\leq k\leq T}(viterbi_i(t_k) \text{ x } P(w_{i+1}|t_j) \text{ x } P(t_j|t_k))$
(8) $previous\_state_{i+1}(t_j):=argmax_{1\leq k\leq T}(viterbi_i(t_k) \text{ x } P(w_{i+1}|t_j) \text{ x } P(t_j|t_k))$
(9) end
(10) end
(11) Comment:Termination and path-readout
(12) $best\_parse_{n+1}:=argmax_{1\leq j\leq T} viterbi_{n+1}(j)$
(13) for j:=n to 1 step -1 do
(14) $best\_parse_j:=previous\_state_{j+1}(best\_parse_{j+1})$
(15) end
(16) $P(best\_parse_{1,...,} best\_parse_n):=max_{1\leq j\leq T} viterbi_{n+1}(t_j)$

Maximum likelihood estimation (MLE): Maximum likelihood is one of the simplest ways to compute probabilities through relative frequencies. In case of HMM, we estimate probability distribution variables for the model parameters $\lambda = (A, B, \pi)$ in the training corpus (Blunsom, 2004; Padró and Padro, 2004), as given below.

$$\pi_i = \frac{C(q_1 = t_i)}{C(q_1)}$$

$$a_{ij} = \frac{C(t_i, t_j)}{C(t_i)}$$

$$b_j(k) = \frac{C(w_k, t_j)}{C(t_j)}$$

Where:

$C(t_i, t_j)$ = Denotes the count (or number of times) that state t_j is followed by state t_i

$C(w_k, t_j)$ = Denotes the number of times w_k tagged with t_j

Smoothing techniques: Essentially MLE is an unstable estimator for statistical inference because of data sparseness (Manning and Schutze, 1999) even for corpus with large number of words. Sparseness means that various numbers of words are infrequent. Therefore in order to overcome this problem it is normally necessary to introduce smoothing techniques, while using with HMM. Different smoothing methods have been proposed and described in the literature and they exhibit different characteristics.

Smoothing is primarily aimed at re-evaluating the probabilities of the words which are quite unlikely to occur and then assign them appropriate non-zero weights. In this way smoothing techniques not just only adjust upward zero probabilities but also downward high probabilities (Chen and Goodman, 1998). This results in resolving the data sparseness issue, a frequently occurring problem. In the following sub-sections we present a brief description of used smoothing methods.

Laplace (add-one) estimation: Laplace estimation is the simple and oldest method for data sparseness and provides us an essential baseline concept about other smoothing techniques that have same parameters. This smoothing method depends on adding one to all the frequency counts. This added value pretends that all the zero probability counts had been seen once in the corpus. We utilized the known approach of adding 1 to seen and unseen events and then added number of word types in Vocabulary (V) to the total number of words (N) in order to keep probability normalized (Jurafsky and Martin, 2000; Padró and Padro, 2004). For unigram model this can be described by:

$$p_{laplace(x)} = \frac{C(x) + 1}{N + V}$$

Lidstone estimation: One noticeable problem with Laplace estimation is that it assigns overvalued probabilities to unseen events, principally not true. This results in erroneous estimation due to incorrect probability values assignment between seen and unseen words. Therefore we try to overcome this problem through Lidstone estimation. This method was introduced by Hardy and Lidstone and can be viewed as a linear interpolation between the MLE estimate and a uniform prior (Manning and Schutze, 1999). In this technique an arbitrary value λ is selected and added to all the events. The value of λ is chosen between 0 and 1 (Padró and Padro, 2004) however the choice of an optimal value is a matter of experimentation. Lidstone estimation for a unigram model is given by:

$$p_{lidstone(x,\lambda)} = \frac{C(x) + \lambda}{N + V\lambda} \quad 0 < \lambda \le 1$$

Expected likelihood estimation: We note that one problem with Lidstone estimation is the choice of an optimal value of λ. Although good guess or experimentation are the ways for the selection of a suitable value of λ yet expectation can be thought of as a theoretically justified choice. As we see that this way one can avoid the problem of assigning a considerably larger weight to unseen events but clearly this choice of λ can not always over perform as that in Lidstone estimation. The Lidstone estimation with a choice of $\lambda = 1/2$ is named as Jeffreys-Perks law, or Expected Likelihood Estimation (ELE) and is given by: (Manning and Schutze, 1999)

$$p_{ELE(x,0.5)} = \frac{C(x) + 0.5}{N + V/2}$$

So, shortly generalized formulation for HMM based Urdu tagger using any of above mentioned smoothing techniques can be written as:

$$\pi_i = \frac{C(s_i(t = 0)) + \lambda}{C(\text{tokens}) + V\lambda}$$

$$a_{ij} = \frac{C(s_i \text{ to } s_j) + \lambda}{C(\text{token}) + V\lambda}$$

$$b_j(k) = \frac{P(s_j \mid w_k)P(w_k)}{P(s_j)}$$

Where:

$$P(s_j) = \frac{C(s_i) + \lambda}{C(\text{tokens}) + V\lambda}$$

$$P(w_k) = \frac{C(w_k) + \lambda}{C(\text{tokens}) + V\lambda}$$

Witten-Bell estimation: This method was illustrated by Witten and Bell. Witten-Bell is one of the simplest estimation yet it has clever assumptions regarding zero frequency events. The basic idea of this method is to count the number of events seen once in training corpus to estimate the number of events unseen. In particular, we say that the probability estimation of the event occurrence for the first time. After assigning the probability of unseen events we have to adjust the extra probability mass. The extra probability mass of events can be adjusted by discounting the probability of all the seen events as shown in the following equations: (Jurafsky and Martin, 2000)

$$P_i^* = \frac{T}{Z(N+T)} \text{ if } (C_i=0)$$

$$P_i^* = \frac{C_i}{N+T} \text{ if } (C_i>0)$$

HMM integration with Witten-Bell smoothing is shown in the following equations.

$$Z = N_{word} * N_{state} - T(token)$$

$$a_{ij} = P(t_j \mid t_i) = \frac{T(t,t_i)}{Z(C(t_i)+T(t,t_i))} \text{ if } (C(t_i,t_j)=0)$$

$$a_{ij} = P(t_j \mid t_i) = \frac{C(t,t_i)}{C(t_i)+T(t,t_i)} \text{ if}(C(t_i,t_j)>0)$$

$$b_j(k) = P(w_k \mid t_j) = \frac{T(w,t_j)}{Z(C(t_j)+T(w,t_j))} \text{ if } (C(w_k,t_j)=0)$$

$$b_j(k) = P(w_k \mid t_j) = \frac{C(w_k,t_j)}{C(t_j)+T(w,t_j)} \text{ if}(C(w_k,t_j)>0)$$

Good-Turing estimation: The Good-Turing (GT) smoothing technique, illustrated by Good and Turing in 1953, provides an estimate for probability mass assignment to zero frequency events based on the probability mass of events occurring once. If Nc refers to the events that occur c times then Good-Turing (GT) estimation can be formalized as (Jurafsky and Martin, 2000):

$$N_c = \sum_{b:c(b)=c} 1$$

$$C^* = (C+1)\frac{N_{c+1}}{N_c}$$

Cut-off value: Additionally, we observed that during training process the frequencies of the rarely occurred words varies from time to time on different training corpus. This observation has great impact of tagging performance. Therefore we introduce cut-off value and this is an optimal value. This value is applied to all the low counts of frequencies because higher counts of frequencies are implicitly reliable. Katz also suggests that in case of English text the cut-off value obtains better result. After introducing the cut-off value we can estimate the frequencies through the following equation: (Feng-Long and Ming-Shing, 2004; Jurafsky and Martin, 2000)

$$c^* = \frac{(c+1)\frac{N_{c+1}}{N_c} - c\frac{(k+1)N_{k+1}}{N_1}}{1-\frac{(k+1)N_{k+1}}{N_1}}, \text{for } 1 \le c \le k$$

HMM integration with Good-Turing smoothing is shown in the following equations.

$$C^*(t_i,t_j) = (C(t_i,t_j)+1)\frac{N_{C(t_i,t_j)+1}}{N_{C(t_i,t_j)}}$$

$$a_{ij}\frac{C^*(t_i,t_j)}{C(t_i)}$$

$$C^*(w_k,t_j) = (C(w_k,t_j)+1)\frac{N_{C(w_k,t_j)+1}}{N_{C(w_k,t_j)}}$$

$$b_j(k) = \frac{C^*(w_k,t_j)}{C(t_j)}$$

The part of speech data and tagset: The data used in our experiments is originated from the EMILLE (Enabling Minority Language Engineering), released on 2004 by Lancaster University. Three major constituents (monolingual, parallel and annotated corpora) are focused in this corpus. The tag set employed to tag the EMILLE corpus consists of 90 tags with some morph-syntactic features (Hardie, 2003).

Experiment and analysis: As usual, experimental results of our tagger have been evaluated through standardize formulation: precision, recall and F-measure (Trommer and Kallulli, 2004) using EMILLE corpora.

$$\text{Precision} = \frac{\text{Correct number of token tag pair occurrence}}{\text{Total number of token tag pair}}$$

$$\text{Recall} = \frac{\text{Correct number of token tag pair occurrence}}{\text{Number of correct token tag pair that is possible}}$$

$$\text{F-measure} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision+Recall}}$$

The results of Hidden Markov Model with different smoothing techniques are shown in Fig. 1. These results show overall precision of different smoothing methods for different size of training corpus. Since evaluation has been carried out using the same information set and under the identical test environment therefore presented comparison provides quite a fair comparison of different smoothing methods. Presented results show that even for sparse events overall accuracy of the tagger improves significantly with the introduction of smoothing techniques. Thus we find that the HMM together with different smoothing methods shows improved results than ordinary Hidden Markov models.

Although an overall tagging accuracy of 90% is achievable using HMM (without smoothing), or by using MLE, yet further improvement of results and data sparseness issue necessitate introduction of some other method. In order to overcome these problems we employ above mentioned smoothing techniques and find significant improvement in results, as shown in Fig. 1. Laplace, Lidstone, Expected, Witten-Bell and Good-Turing smoothing methods estimated the parameters required for the data sparseness to increase the overall accuracy from 83.38 to 92.80%, from 87.49 to 95.50%, from 85.04 to 93.90%, from 86.28 to 95.17 and from 88.15 to 96.00%, respectively. In our experimentation we found that Good-Turing and Lidstone smoothing methods achieved best accuracy results. If we consider overall accuracy as the definitive indicator of efficiency, Lidstone method is certainly better than the well-define Witten-Bell smoothing method.

Table 2 presents inclusive comparison of overall accuracy, known word accuracy, recall rate and F-measure for different smoothing methods. Since Urdu is free word language and has relatively complex morphology than English, therefore unknown word accuracy for Urdu is not so high. For the case of unknown words handling Fig. 2 shows that the Witten-Bell method has good adaptability than other smoothing methods.

This is due to the fact that Witten-Bell method assigns better probabilities to unknown words and has more inclusive parameters for handling unknown word as compared to other smoothing methods. While Witten-Bell smoothing yields best results for unknown words, Lidstone smoothing can be ranked as the second best technique. This is specifically due to the presence of flexible parameter λ that plays important role in redistribution of the probability values.

Evaluation: In Table 3, performance of different smoothing methods and for different number of words has been tested. Presented evaluation depends on two null hypotheses. While one correspond to the situation that all the number of word means are equal, the other corresponds to the situation that accuracy means of all the methods are equal. We test these hypotheses by applying two-way ANOVA and compute following estimates of the population variance (Chaudhary and Akmal, 1998).

Table 2: Comprehensive comparison of different smoothing techniques for Urdu POS

Methods	Overall accuracy	Known word accuracy	Recall	F-Measure
Maximum likelihood	90.00	93.91	88.32	89.16
Laplace	92.80	94.60	91.06	91.93
Lidstone	95.50	96.70	94.21	94.86
Expected likelihood	93.90	95.50	92.02	92.96
Witten-Bell	95.17	96.10	93.00	94.09
Good-Turing	96.00	97.80	95.10	95.55

Table 3: Two-way ANOVA for different smoothing methods and number of words

Source of variation	df	Sum of squares	Mean square	F	p
Between smoothing methods	5	457.86	91.571	136.95	0.000
Between number of words	4	2515.22	628.806	940.41	0.000
Error	20	13.37	0.669		
Total	29	2986.45			

$S = 0.8177$, $R^2 = 99.55\%$, $R^2(adj) = 99.35\%$

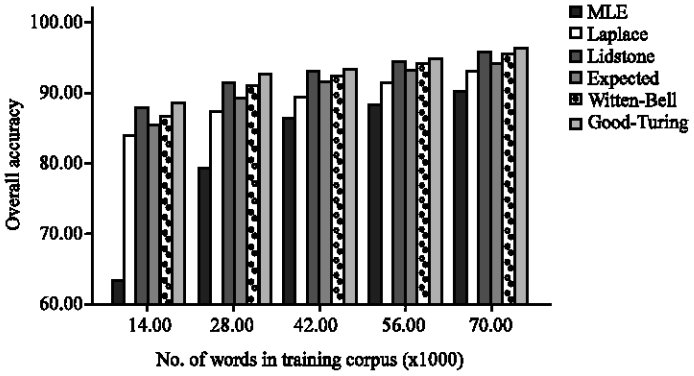

Fig. 1: Learning bars for overall accuracy (MLE, Laplace, Lidstone, Expected, Witten-Bell and Good-Turing)

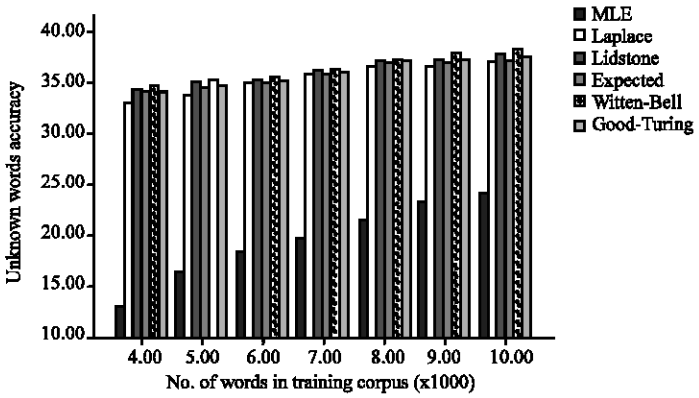

Fig. 2: Learning bars for unknown words accuracy (MLE, Laplace, Lidstone, Expected, Witten-Bell and Good-Turing)

Therefore, the hypothesis test of number of words means are equal, we calculate as:

$F_1 = \frac{s_1^2}{s_3^2}$ at 0.05 level of significance provided $F_1 \geq F_{0.05}$; [(r-1), (r-1) (c-1)]

Similarly, the hypothesis test of all the methods are equal, we calculate as:

$F_2 = \frac{s_2^2}{s_3^2}$ at 0. 05 level of significance provided $F_2 \geq F_{0.05}$; [(c-1), (r-1) (c-1)]

These hypotheses are dependent on the results shown in Fig. 1 and are important for the evaluation of our tagger behavior. We notice that different smoothing techniques exhibit different behaviors due to the reason that all methods have their own characteristics and computing parameters. However the choice of best depends on several factors and still varies from situation to situation. Table 3 show that F = 136.95 indicating higher value than tabular F-value for this reason the differences among the accuracy using different smoothing method are significant. Furthermore it is also evident form Table 4 that results differ considerably for varying number of words and continue to improve with the increasing size of corpora. Since F = 940.14 has higher value than tabular F-value with 4 and 20 degree of freedom. We conclude that the differences in the effectiveness of varying number of words are significant.

Error analysis: The main task of part of speech tagging is to resolve the syntactic category ambiguities. Though in most cases tagger successfully resolves these ambiguities but some of the cases are relatively hard to resolve (Kristina and Christopher, 2000). The confusion matrix, Table 4, consists of (n*n) entries where rows contain accurate tag and columns indicate the tagger assigned tags. Every element of the matrix contains the number of times tag i was categorized as tag j (Bird *et al.*, 2007). Our error analysis matrix contains the most confusing tag pairs whereas Table 5 contains the tags which are most inflected with other grammatical categories.

Using different smoothing methods with HMM we find that the most confusing tags appear to be almost the same, as presented in the confusion matrix Table 4. The off diagonal entries of the confusion matrix give misclassification between different parts of speech. For instance the (JJU, Noun) cell entry value 979 indicates the number of words which were JJU's but wrongly tagged as Noun. Careful observation of Table 4 results in deducing following important points about Urdu part of speech tagging error.

- The confusion matrix shows that the occurrences of misclassification cells are frequently based on nouns, verbs, adjectives, pronouns and numeral tags. However, since noun tags appear comparatively more frequently in the corpus therefore the frequency distribution of the noun tags are dominate over other tags.
- Urdu is partially word-free language. In our experiment, we used partially morph-syntactic tagset as well as some morph-syntactic tags are harder to define because of morphological complexity of the Urdu language. Therefore internal level misclassification of tags is harder to resolve.
- Urdu is a highly inflected language and since several grammatical categories of inflections are very closely related, as described in Table 5, therefore misclassification problem may often be encountered. For example, as noun and adjective pair of tags are strongly inflected among each other therefore misclassification between these two pair of tags are quite higher than the other pair of tags, also evident from Table 4.

Table 4: Confusion matrix for most confusion pairs overall

	Tagger assigned tags											
Correct tags	Noun	Verb	JJU	JDNU	RR	JD	OrdN	ProP	ProD	ProI	ProR	Total
Noun	-	214	391	51	5	0	0	112	14	1	2	790
Verb	317	-	113	7	0	3	4	16	1	0	1	462
JJU	979	147	-	43	6	0	0	83	11	0	4	1273
JDNU	41	2	12	-	0	0	0	4	0	0	0	59
RR	14	1	0	0	-	0	0	1	0	0	0	16
JD	15	5	0	0	0	-	0	0	0	0	0	20
OrdN	17	0	21	2	0	0	-	3	0	0	0	43
ProP	13	0	1	0	0	0	0	-	0	0	0	14
ProD	7	2	3	0	0	0	0	0	-	0	0	12
ProI	3	0	0	0	0	0	0	0	0	-	0	3
ProR	11	0	0	0	0	0	0	0	0	0	-	11
Total	1417	371	541	103	11	3	4	219	26	1	7	2703

Table 5: Urdu language inflected part of speech tags and grammatical categories

POS	Gender	Case	Number	Person	Aspect	Mood	Degree	Tense
Noun		X	X					
Verb	X		X		X	X		X
Adjective	X	X	X				X	
Pronoun	X	X	X	X				

*: X = liaison between inflected parts of speech tags and grammatical categories

CONCLUSION AND FUTURE WORK

In this study, we proposed the initial implementation of HMM to one of the partially free word and morphology rich language Urdu. During experimental results we note that the general HMM based method doesn't perform well. Due to this reason we combined different smoothing techniques with HMM to handle data sparseness problem. However, we also note that after the combination of smoothing methods the distribution of the probability are well distributed among data. Because of this the overall accuracy and unknown word accuracy is significantly increased. For the evaluation of our experiment results we selected two assumptions: one different size of training corpus and other different smoothing methods through different standardize formulation. The Significance of our results is estimated via applying ANOVA method.

In future, we intend to develop novel methods to improve overall accuracy and specifically unknown words in Urdu and other word-free languages. We aim to find out ways to improve the language model behavior without increasing the training corpus and by integrating linguistics knowledge

REFERENCES

Bird, S., E. Klein and E. Loper, 2007. Natural language processing in python. University of Pennsylvania, nltk. sourceforge. net/index. php/Book.

Blunsom, P., 2004. Hidden Markov Models. Technical Report. www. cs. mu. oz. au/460/2004/materials/hmm-tutorial. pdf.

Chaudhary, S.M. and S. Akmal, 1998. Introduction to Statistical Theory. Markazi Kutub Khana, Lahore.

Chen, S.F. and J. Goodman, 1998. An empirical study of smoothing techniques for language modeling. Technical report center for research in computing technology. Harvard University. Cambridge, Massachusetts, pp: 359-394.

Fatima, Al.S. and A. Guessoum, 2006. A hidden markov model-based POS tagger for arabic. In: Proceeding of the 8th International Conference on the Statistical Analysis of Textual Data, France, pp: 31-42.

Feng-Long, H. and Yu. Ming-Shing, 2004. Study on Good-Turing and a novel smoothing method based on real corpora for language models. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Singapore, pp: 3741-3745.

Hardie, A., 2003. Developing a tagset for automated part-of-speech tagging in Urdu Corpus. In: Proceeding of the Linguistics conference. Department of Linguistics, Lancaster University.

Javed, I., 1985. New Urdu Grammar. Advance Urdu Burew. New Dehali.

Jurafsky, D. and J.H. Martin, 2000. SPEECH and LANGUAGE PROCESSING: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice-Hall.

Kristina, T. and M. Christopher, 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In Proceeding of the Joint SIGDAT Conference EMNLP/VLC, pp: 63-70.

Manning, C. and H. Schütze, 1999. Foundations of Statistical Natural Language Processing. MIT Press. Cambridge.

Padró, M. and L. Padró, 2004. Developing competitive HMM PoS taggers using small training corpora. In: Proceedings of the 4th International Conference, EsTAL, Spain, pp: 127-136.

Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proceeding of the IEEE, pp: 257-286.

Samuelsson, C., 1996. Handling sparse data by successive abstraction. In: Proceeding of the 16th Conference on Computational Linguistics, pp: 895-900.

Scott, M.T. and M.P. Harper, 1999. A second-order Hidden Markov Model for part-of-speech tagging. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, pp: 175-182.

Sigletos, G., G. Paliouras and V. Karkaletsis, 2002. Role identification from free text using Hidden Markov Models. Proceedings of the Second Hellenic Conference on AI: Methods and Applications of Artificial Intelligence, pp: 167-178.

Trommer, J. and D. Kallulli, 2004. A morphological tagger for standard Albanian. In Proceedings of LREC, Portugal.