HMM Based POS Tagging System for 8 Different Languages and Several Tagsets

Dr. Ahmed Hussein Aliwy
Computer Science Department, University of Technology/Baghdad.
Email:Ahmed_7425@yahoo.com
Rosual Ali Radie
Computer Science Department, University of Technology/Baghdad.
Hiba Sarteel Hamed
Computer Science Department, University of Technology/Baghdad.

Revised on:9/1/2014 & Accepted on: 13/5/2014

ABSTRACT

We propose, in this paper, Part-Of-Speech (POS) tagging system is proposed which based on Hidden Markov Model (HMM) for several languages. HMM is implemented using Viterbi algorithm on 8 languages; English, Hindi, Telugu, Bangla (Bengali), Marathi, Standard Chinese, Portuguese and Spanish. The data for these languages were taken from the freely available corpora: Brown, NPS-Chat, Indiana, Sinica, Floresta and CESS-ESP Corpora.

HMM is the most learning method used in many NLP applications, especially POS tagging. HMM tagger was implemented by other researchers for a lot of languages, where each one take his mother tongue language.

system testing is done by splitting each corpus to 99% training and 1% testing. This test is repeated for 10 times by changing the training and test data. The accuracies (average for all 10 tests) for English (using two tagsets of 40 tags and 472 tags), English (NPS corpus), Hindi, Telugu, Bangla or Bengali, Marathi, Standard Chinese, Portuguese (using two tagsets of 32 tags and 269 tags), and Spanish (using two tagsets of 14 tags and 289 tags) are (95.3%& 92.39%), 87.17%, 81.3%, 74.03%, 72.01%, 69.56%, 87.59%, (84.56%& 83.95%), and (94.26%& 92.08%) respectively.

Several languages are taken for recording the limitations of HMM tagger on different languages as will be seen, I.e, the limitations of using one method on many different languages are recorded. Same corpus annotated with different tagsets is taken for studying the effect of tagset's size. Also two different corpora, for the same language, are taken. According to our knowledge, there isn't study implemented HMM on such various cases as in our work.

We provide an executable application[1] for tagging all words in any sentence for any of the used 8 languages in our work. The unknown words (words not exist in the trained data) are manipulated by a simple method as Laplace smoothing.

Keywords: HMM tagger, multi-language tagger.

[1]This application are done only for reviewers (review process) and not permitted to any one for distributing it for any reason because the copyright are reserved to us.

نظام ترميز اقسام الكلام معتمداموديل ماركوف المخفي لثمان لغات وعدة مجاميع ترميز

الخلاصة

نقترح في بحثنا نظام ترميز الكلمات باقسام الكلام باستخدام طريقة HMM لعدة لغات.طبّقناHMM باستخدام خوارزمية Viterbi على ثمان لغات هي اللغة الانجليزية والهندية والتلوكو والبنكالية والمهاراتية والصينية القياسية والبرتغالية والاسبانية. البيانات لهذه اللغات اخذناها من ذخائر (مدونات) موجودة بشكل مجاني وهيNPS-Chat Indiana,Brown, , Floresta ,Sinica و CESS-ESP.

HMM هي من اكثر طرق التعلم المستخدمة في تطبيقات كثيرة لمعالجة اللغات الطبيعية خصوصا الترميز باقسام الكلام,وان بعض الباحثين الاخرين نفذوا مرمز HMM على لغات كثيرة حيث كل باحث نفذها على لغته. تنفيذنا للنظام تم من خلال تقسيم كل ذخيرة (البيانات) الى 99% للتدريب و 1% للفحص, هذه العملية تعاد لعشرة مرات من خلال تغيير بيانات التدريب والفحص, وكانت الدقة (كمعدل لجميع الفحوصات)للغة الانجليزية (مجموعتي ترميز 40 و472 رمز) والانكليزية (ذخيرة NPS-Chat) والهندية والتلوكو والبنكالية والصينية القياسية والبرتغالية (مجموعتي ترميز 32 و269 رمز) والاسبانية (مجموعتي ترميز 14 و289 رمز) هي (95.3% و 92.39%), 87.17%, 81.3%, 74.03%, 72.01%, 69.56%, 87.59%, (84.56% و 83.95%) و (94.26%, 92.08%) على الترتيب.

اللغات المختلفة اخذناها لغرض تسجيل تحديدات مرمز HMM على لغات مختلفة كما سنرى, وهذا يعني تسجيل التحديدات باستخدام طريقة واحدة على عدة لغات. كذلك اخذنا نفس الذخيرة معنونة بمجموعة رموز مختلفة لغرض دراسة تاثير حجم مجموعة الرموز,بالاضافة الى ذلك اخذنا ذخيرتين مختلفتين لنفس اللغة,فحسب معلوماتنا ليس هناك دراسة معمقة منفذة على مرمز HMM بنفس الحالات المأخوذة في هذا العمل.

وفرنا ايضا برنامج تطبيقي لترميز جميع الكلمات لاي جملة من اي من اللغات المستخدمة في عملنا. الكلمات الغير معروفة (غير موجودة في بيانات التدريب) عالجناها بطريقة بسيطة جدا وهي Laplace smoothing.

INTRODUCTION

POS tagging is the most studied field in natural language processing (NLP) area. It is very important task for many NLP applications such as machine translation (MT) and many others. POS tagging, or simply tagging, is the process of classifying words into their parts-of-speech and labeling them accordingly[1].

In such task, we are given some observation(s) and our job is to determine which of a set of classes it belongs to. Part-of-speech tagging is generally treated as a sequence classification task. So here the observation is a sequence of words (may be sentence), and it is our job to assign them a sequence of part-of-speech tags[2].

For understanding tagging problem, suppose we try to classify (tagging) a sequence of words $w_1 \ldots w_n$by a set of classes (tags) $\{t_1 \ldots t_m\}$. What is the best sequence of classes (tags) which corresponds to this sequence of words? The Bayesian interpretation of this task starts by considering all possible sequences of classes (in this case, all possible sequences of tags). Out of this universe of tag sequences, we want to choose the tag sequence which is most probable given the observation sequence of these n words[2].

A part-of-speech (POS) tagger assigns a POS label to each word of an input text. The tagger first obtains the set of possible POS tags for each word from a lexicon and then disambiguates between them based on the word context[3]. Parts-of-speech are also known as word classes or lexical categories. The collection of tags used for a particular task is known as a tagset[1].

There are many approaches used for tagging, one of them Hidden Markov Model (HMM). HMM used for tagging complete sentence according to the context. In this work we will implement a HMM tagger on several languages with several tests.

Different corpora for the same language are used and same corpus, annotated by different tagsets, is also used. Finally, executable application will be provided which used for tagging any input (sentence) from any used language.

HMM on tagging

Often we want to consider a sequence of random variables that aren't independent, but rather the value of each variable depends on previous elements in the sequence. For many such systems, it seems reasonable to assume that all we need to predict the future random variables is the value of the present random variable, and we don't need to know the values of all the past random variables in the sequence. This is called Markov Model. In an HMM, the state sequence that the model passes through is not known, but only some probabilistic function of it[4].

Use of a Hidden Markov Model to do part-of-speech-tagging is a special case of Bayesian inference. Bayesian inference or Bayesian classification was applied successfully to many language problems[2].

Hidden Markov Model (HMM) is the most frequently used technique for POS tagging. It can be used for tagging one complete sentence at a time, by selecting the most likely sequence of tags for its word [5].It uses the formula[2]:

$$t_1^n = \arg\max_{t_1^n} \quad p(t_1^n \mid w_1^n) \qquad \ldots(1)$$

$p(t_1^n \mid w_1^n)$ is the probability of tags sequence $t_1 \ldots t_n$ given that the words sequence $w_1 \ldots w_n$. t_1^n is the best tags sequence for given words where the $p(t_1^n \mid w_1^n)$ maximum. Equation 1 can not be computed directly, therefore by using Bayes' rule it will be[2]:

$$\hat{t}_1^n = \arg\max_{t_1^n} \quad p(w_1^n \mid t_1^n)\, p(t_1^n) \qquad \ldots(2)$$

HMM tagger simplifies this formula by two assumptions. The first assumption is that the probability of a word depends on its part-of-speech tag and is independent of other words around it, and of the other tags around it[2]:

$$p(w_1^n \mid t_1^n) \approx \prod_{i=1}^{n} p(w_i \mid t_i) \qquad \ldots(3)$$

The second assumption is that the probability of a tag appearing depends only on the previous tag, the bigram assumption[2]:

$$p(t_1^n) \approx \prod_{i=1}^{n} p(t_i \mid t_{i-1}) \qquad \ldots(4)$$

From equations 3 & 4, we will get:

$$\hat{t}_1^n = \arg\max_{t_1^n} \quad p(t_1^n \mid w_1^n) \approx \arg\max_{t_1^n} \prod_{i=1}^{n} p(w_i \mid t_i)\, p(t_i \mid t_{i-1}) \qquad \ldots(5)$$

This is the first order HMM. The second order HMM uses trigram assumption where the current tag depends on the two previous tags only.

$$\hat{t}_1^n = \arg\max_{t_1^n} p(t_1^n \mid w_1^n) \approx \arg\max_{t_1^n} \prod_{i=1}^{n} p(w_i \mid t_i)\, p(t_i \mid t_{i-2} t_{i-1}) \quad \ldots(6)$$

These parameters are estimated from training on annotated corpus as follows:

$$p(t_i \mid t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})} \quad \ldots(7)$$

$C(t_{i-1}, t_i)$ is a counts of (t_{i-1}, t_i) appearing in the training data. The important thing here, we must know that t_{i-1} & t_i are variables for all tags in the tagset not only one tag.

$$p(w_i \mid t_i) = \frac{C(t_i, w_i)}{C(t_i)} \quad \ldots(8)$$

$C(t_i, w_i)$ is counts of the word w_i appears with tag t_i in the training data.

The Used Languages and Corpora

Several well-known languages are used, in our work, as English (using Brown corpus & NPS-Chat), Hindi (using Indiana corpus), Bangla (using Indiana corpus), Marathi (using Indiana corpus), Telugu (using Indiana corpus), Chinese (using Sinica corpus), Portuguese (using Floresta Corpus) and Spanish (using CESS-ESP Corpus). In the implementation, we take two tagsets For English, Portuguese and Spanish languages.

English(Brown corpus) is a simple inflected language comparing with the other languages. It is now the most widely used language in the world. Brown and NPS-Chat corpora are taken, in our work, which are freely available. Brown corpus contains57340 sentences (1161192 tagged tokens ≈ Million words). Two tagsetsare used in Brown Corpus which are taken in our test. NPS-Chat corpus contains10567 sentences (45010 tagged tokens).

Hindi (Indiana corpus) is standardized of the Hindustani language. Hindi is one of the official languages of India. Indiana corpus is used, in our work, which is freely available containing 3631sentences (49365 tagged token) for four languages. It contains 541sentences of Hindi language (9475 tagged tokens).

Telugu (Indiana corpus) is an official language in some position in India. Telugu ranks third by the number of native speakers in India (74 million speakers). Indiana corpus contains 994sentences of Telugu language (10004tagged tokens).

Bangla or Bengali(Indiana corpus) is native to Bangladesh, the Indian state of West Bangladesh, and parts of the Indian states. Bengali is one of the most spoken languages, ranked seventh in the world(250 million speakers).Indiana corpus contains 899 sentences of Bangla language (10427 tagged tokens).

Marathi language (Indiana corpus) is the official language of Maharashtra state of India. Marathi has the fourth largest number of native speakers in India(73 million

speakers).Indiana corpus contains 1197 sentences of Marathi language (19459 tagged tokens).

Standard Chinese (Sinica corpus) is a standardized variety of Chinese. It is the sole official language Republic of China. Sinicacropus designed for analyzing modern Chinese. Every text in the corpus was segmented. Part from it is freely available containing 9999sentences of Standard Chinese language (91627 tagged tokens).

Portuguese is official language of little countries as Portugal, Brazil, and others. Floresta corpus is a publicly available Treebank for Portuguese language. It contains9266 sentences of Portuguese language(211852 tagged tokens). Two tagsets are used in Floresta corpus and are taken in our test.

Spanish (CESS-ESP Corpus) is official language of of Spain (406 million speakers). It is one of the six official languages of the United Nations. CESS-ESP Corpus is part of CESS-ECE project. It contains 6030 sentences of Spanish language (192685 tagged tokens). Two tagsets are used in CESS-ESP corpus and they are taken in our test.

Related Works

There are many POS tagging works on a lot of languages, some of them used HMM with private language, but we can't list them here for the paper limit. We list the works which has the same approach on the same language and/or the same corpus.

Avinesh and Karthik[6]used CRF(Conditional random field) and TBL (Transformation-based learning) based POS tagger and has an accuracy of about 77.37%, 78.66%, and 76.08% for Telugu, Hindi and Bengali languages respectively. They used Indian corpus, the same corpus used in our work. The size of data used by them was much more than these available to us. The tagset is the same in both works.

Singh et Al.[7] used Trigram Method for tagger development on Marathi language. It is second order HMM. They used a private test corpus of 2000 sentences (48,635 words). They used IL POS tagset which consists of 24 tags. The accuracy of the system was 91.63%.

Nisheeth et Al.[8]used HMM on Hindi Language. They used IL POS tagset and achieved an accuracy of 92% on a corpus of 15,200 sentences (358288 words).

Rodrigues et Al. [9] combined HMMs and character language models which were applied to Portuguese texts. In this approach, the emission probabilities for each hidden state in a HMM are estimated by a proper character language model. The tagger built has been trained and tested on Bosque, a subset of Floresta Treebank. They reached 96.2% accuracy with a tagset of 39 tags and 92.0% with a tagset of 257 tags.

Chao-hung & Cheng-Der[10] used first order HMM tagger on Chinese language with word identification possibility. They achieved an accuracy of 96% on a private corpus.

Padró &Padró[11] used tri-grams and quad-grams HMM tagger on Spanish language. They achieved 96.90% and 96.73% accuracies for trigram and quad-gram using Linear Interpolation. They achieved 96.85% and 96.22%accuracies for trigram and 4-gram respectively using Lidstone's law.

Our work different than the other works by the following nodes:

1- Applying one approach on many languages in order to record the behavior of this approach on these languages.

2- Very different languages are selected from the world languages in order to record the possibility of getting high accuracy on tagging for the same approach.
3- More than one tagsets, for the same corpus, are taken. It is useful to record how tagset size affects the results.
4- Different corpora, for the same language, are used.
In summary various testing conditions are reported which make novelty of our work comparing with related works.

Implementation and Results

The used data are partitioned, for each corpus, into 100 parts. 99% is taken as training and 1% as test. These data partitions can be 100-fold-cross-validation with one difference[2] where the test is repeated for 10 times not for 100 times (see Figure 1 for more details on partitioning). The samples (test data) are very small because some of the used corpora are very small which lead to many unknown words then raising the errors.

The used data set are 6 corpora: Brown, NPS-Chat, Indiana, Sinica, Floresta and CESS-ESP Corpora. Each corpus contains one language except Indiana corpus which contains 4 languages. I.e. the used languages, in our work, are 8 languages: English, Hindi, Telugu, Bangla or Bengali, Marathi, Standard Chinese, Portuguese and Spanish.

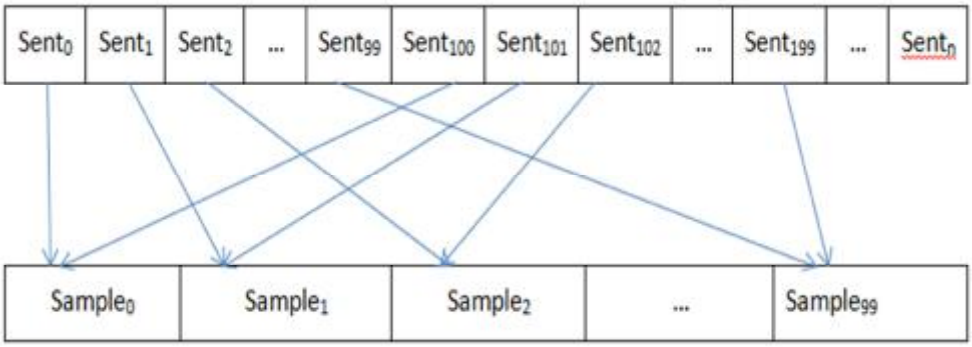

Figure (1): partitioning a corpus to100 equaled size subsamples.

First order HMM tagger was implemented using Viterbi algorithm. We used Laplace smoothing for sparse data and unknown words.
Two corpora, for English language, are used with three tagsets. These corpora are Brown corpus and NPS-Chat corpus. Brown corpus has got annotation with two tagsets of 40 tags and 472 tags. The results of implementing HMM tagger on Brown corpus are shown in Tables1&2 respectively. Implementing HMM tagger on NPS-Chat corpus is shown in Table3.

[2]In k-fold cross-validation, the original sample is randomly partitioned into k equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k−1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data.

The results of Implementing HMM tagger on Hindi, Telugu, Bangla and Marathi languages are shown in Tables4, 5, 6, and 7 respectively. These languages are taken from Indiana corpus.

Sinica corpus, for Standard Chinese language, is used. The results of implementing HMM on Standard Chinese language is shown in Table8.

Floresta corpus, for Portuguese language, is used with two tagsets. It has got annotation with a tagset of 32 tags and a tagset of 269 tags. The results of implementing HMM tagger on Floresta corpus are shown in Tables9&10 respectively.

CESS-ESP corpus, for Spanish language, is used with two tagsets. It has annotation with a tagset of 32 tags and a tagset of 269 tags. The results of implementing HMM tagger on CESS-ESP corpus are shown in Tables11& 12 respectively.

Table (1): Running HMM on English using Brown corpus with a tagset of 32 tags.

No. of test sample	Trained data size	Tested data size	Number of match	Number of wrong	Accuracy
0	1149122	12070	11485	585	0.951
10	1148818	12374	11743	631	0.949
20	1149140	12052	11505	547	0.954
30	1149976	11216	10715	501	0.955
40	1149447	11745	11234	511	0.956
50	1149398	11794	11262	532	0.954
60	1149704	11488	10937	551	0.952
70	1149196	11996	11455	541	0.954
80	1150001	11191	10668	523	0.953
90	1149839	11353	10813	540	0.952

Table (2): Running HMM on English using Brown corpus with a tagset of 472 tags.

No. of test sample	Trained data size	Tested data size	Number of match	Number of wrong	Accuracy
0	1149122	12070	10979	1091	0.91
10	1148818	12374	11257	1117	0.91
20	1149140	12052	11052	1000	0.917
30	1149976	11216	10311	905	0.919
40	1149447	11745	11100	645	0.945
50	1149398	11794	10836	958	0.919
60	1149704	11488	10591	897	0.922
70	1149196	11996	11101	895	0.925
80	1150001	11191	10391	800	0.929
90	1149839	11353	10708	645	0.943

Table (3): Running HMM on English using NPS Chat corpus

No. of test sample	Trained data size	Tested data size	Number of match	Number of wrong	Accuracy
0	44507	503	434	69	0.863
10	44502	508	460	48	0.906
20	44519	491	376	115	0.766
30	44490	520	459	61	0.883
40	44543	467	424	43	0.908
50	44575	435	396	39	0.91
60	44630	380	340	40	0.895
70	44577	433	376	57	0.868
80	44548	462	395	67	0.855
90	44507	503	434	69	0.863

Table (4): Running HMM on Hindi language using Indiana corpus

No. of test sample	Trained data size	Tested data size	Number of match	Number of wrong	Accuracy
0	9356	119	104	15	0.873
10	9363	112	89	23	0.794
20	9423	52	43	9	0.826
30	9363	112	95	17	0.848
40	9375	100	83	17	0.83
50	9398	77	60	17	0.779
60	9395	80	52	28	0.65
70	9374	101	89	12	0.881
80	9379	96	77	19	0.802
90	9403	72	61	11	0.847

Table (5): Running HMM on Telugu language using Indiana corpus

No. of test sample	Trained data size	Tested data size	Number of match	Number of wrong	Accuracy
0	9918	86	67	19	0.779
10	9876	128	94	34	0.734
20	9899	105	75	30	0.714
30	9910	94	68	26	0.723
40	9891	113	87	26	0.769
50	9894	110	78	32	0.709
60	9924	80	60	20	0.75
70	9902	102	71	31	0.696
80	9919	85	62	23	0.729
90	9909	95	76	19	0.80

Table (6): Running HMM on Bangla language using Indiana corpus

No. of test sample	Trained data size	Tested data size	Number of match	Number of wrong	Accuracy
0	10300	127	109	18	0.858
10	10308	119	96	23	0.806
20	10347	80	58	22	0.725
30	10329	98	36	62	0.367
40	10337	90	29	61	0.322
50	10345	82	67	15	0.817
60	10331	96	75	21	0.781
70	10315	112	96	16	0.857
80	10303	124	104	20	0.838
90	10327	100	83	17	0.83

Table (7): Running HMM on Marathi language using Indiana corpus

No. of test sample	Trained data size	Tested data size	Number of match	Number of wrong	Accuracy
0	19222	237	54	183	0.227
10	19263	196	155	41	0.790
20	19271	188	38	150	0.202
30	19247	212	167	45	0.787
40	19230	229	186	43	0.812
50	19247	212	179	33	0.844
60	19269	190	159	31	0.836
70	19264	195	156	39	0.8
80	19321	138	106	32	0.768
90	19276	183	163	20	0.890

Table (8): Running HMM on Chinese language using Sinica corpus

No. of test sample	Trained data size	Tested data size	Number of match	Number of wrong	Accuracy
0	90693	934	824	110	0.882
10	90712	915	793	122	0.866
20	90718	909	798	111	0.877
30	90734	893	792	101	0.886
40	90724	903	787	116	0.871
50	90733	894	778	116	0.870
60	90684	943	822	121	0.871
70	90715	912	808	104	0.885
80	90662	965	844	121	0.874
90	90712	915	803	112	0.877

Table (9): Running HMM on Portuguese language using Florestacorpuswith a tagset of 32 tags.

No. of test sample	Trained data size	Tested data size	Number of match	Number of wrong	Accuracy
0	209650	2202	1866	336	0.847
10	209802	2050	1734	316	0.845
20	209890	1962	1645	317	0.838
30	209708	2144	1809	335	0.843
40	209876	1976	1659	317	0.839
50	209808	2044	1739	305	0.850
60	209860	1992	1671	321	0.838
70	209817	2035	1751	284	0.860
80	209693	2159	1846	313	0.855
90	209782	2070	1742	328	0.841

Table (10): Running HMM on Portuguese language using Floresta corpus with a tagset of 269 tags.

No. of test sample	Trained data size	Tested data size	Number of match	Number of wrong	Accuracy
0	209650	2202	1890	312	0.858
10	209802	2050	1762	288	0.86
20	209890	1962	1487	475	0.758
30	209708	2144	1826	318	0.852
40	209876	1976	1699	277	0.86
50	209808	2044	1744	300	0.853
60	209860	1992	1710	282	0.858
70	209817	2035	1741	294	0.856
80	209693	2159	1717	442	0.795
90	209782	2070	1750	320	0.845

Table (11): Running HMM on Spanish language using CESS-ESP corpus with a tagset of 14 tags.

No. of test sample	Trained data size	Tested data size	Number of match	Number of wrong	Accuracy
0	190755	1930	1814	116	0.939
10	190546	2139	2000	139	0.935
20	190661	2024	1921	103	0.949
30	190871	1814	1711	103	0.943
40	190699	1986	1878	108	0.945
50	190972	1713	1616	97	0.943
60	190702	1983	1877	106	0.946
70	190825	1860	1743	117	0.937
80	190887	1798	1703	95	0.947
90	190913	1772	1670	102	0.942

Table (12): Running HMM on Spanish language using CESS-ESPcorpuswith a tagset of 289 tags.

No. of test sample	Trained data size	Tested data size	Number of match	Number of wrong	Accuracy
0	190755	1930	1761	169	0.912
10	190546	2139	1982	157	0.927
20	190661	2024	1874	150	0.926
30	190871	1814	1680	134	0.926
40	190699	1986	1837	149	0.925
50	190972	1713	1596	117	0.932
60	190702	1983	1826	157	0.921
70	190825	1860	1712	148	0.920
80	190887	1798	1601	197	0.890
90	190913	1772	1646	126	0.929

Discussion and Future Work

As we see, our work took many tests on several languages using several annotated corpora. Our test focused on some aspects: (i) different languages, (ii) different tagsets for same corpus (same language), (iii) different corpora for the same language. There are huge differences for the results in Tables1 to 12. There are many reasons for these differences which can be summarized by three nodes: (i) nature of language, (ii) size of tagset and (iii) size of training data:

Nature of The Language

there is a huge difference among the morphological features of the used languages. In turn there are differences in the complexity of these languages. In turn we need to large training data in case of rich inflected languages. This interprets why, for the same size of training data for two different languages, the results are different. For example, the accuracy (average of all 10 tests) of Marathi language is 69.56% in spite of the size of the trained data is 19k token but the accuracy (average) of Hindi language is 81.3% in spite of the size of trained data is half of the size of Marathi language (see Tables4&8).

Size of Tagset

we used, for English language in our test, a corpus annotated by a tagset of 40 tags and a tagset of 472 tags. Tables1&2 show the results using Brown corpus annotated using these two tagsets respectively. Easily, we can see that the results are dropped down when we used large tagset. This result is logical and expected because large tagset means more information about the words then more ambiguity which leads to more errors and then low accuracy. But, it is not general rule as we will see. Floresta and CESS-ESP corpora are also annotated with two tagsets. Table9&10 show the results of using Floresta corpus using tagsets of 32 tags and 269 tags. We can see that, in most tests, the accuracies in Table 9 are less than the accuracies in Table 10 in spite of the tagset is smaller. Tables11&12 show the results of using CESS-ESP corpus using tagsets of 14 tags and 289 tags.

Size of Training Data

If we used small data set, there are many words from the language not existing in this data set. These words are unknown words. The unknown words are highly affected the accuracy of the system. Also, large data set will give good statistics for learner. Surely, increasing the size of learning data will increase the accuracy. See the results in Tables1&3.

There are some tests have dropped down in the accuracy as(36.7%&32.2%) in Table 6 and the accuracies (22.7% & 20.2%) in Table7. They are very small accuracies. After analyzing the used data in training and test samples manually, we found there are many unknown words in same sentence. This will lead to more errors collected from the context then drop down the accuracy.

We suggest, as future, using trigram and quad-gram HMM tagger on the same test. other taggers as Brill and Maximum entropy tagger can be used for the same data. Unknown words are manipulated by using more efficient approach than Laplace smoothing.

References:

[1] S. Bird, E. Klein, and E. Loper: Natural Language Processing with Python. Book, Published by O'Reilly Media, 2009.

[2] D. Jurafsky& J. Martin: "Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition". Prentice Hall, 2008.

[3] C. Alexander, F. Chris, and L. Shalom "The Handbook of Computational Linguistics and Natural Language Processing" (by) John Wiley & Sons, Ltd., Publication UK, 2010.

[4] C. Manning and H. Schütze: Foundations of Statistical Natural Language Processing. MIT Press. Cambridge, London, 1999.

[5] A. Aliwy:Arabic Morphosyntactic Raw Text Part of Speech Tagging SystemPhD Dissertation, University of Warsaw, Poland, 2013.

[6] P. Avinesh and G. Karthik: Part-Of-Speech Tagging and Chunking using Conditional Random Fields and Transformation Based Learning.In Proceeding of SPSAL2007 conference, IJCAI, India, 2007.

[7] J. Singh, N. Joshi, I. Mathur, "Part of Speech Tagging of Marathi Text Using Trigram Method", International Journal of Advanced Information Technology, pp 35-41, Vol 3. No. 2, 2013.

[8] J. Nisheeth , D. Hemant, M. Iti, "HMM Based POS Tagger for Hindi", Proceedings of 2nd International Conference on Artificial Intelligence and Soft Computing, Organized by AIRCC, India. Published in Advances in Intelligent and Soft Computing Series, Springer Verlang, 2013.

[9] M. Rodrigues, H. maia and G. Bonorino: Part-of-Speech Tagging of Portuguese Using Hidden Markov Models with Character Language Model Emissions. Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology, pages 159-163, Cuiaba, Brazil, 2011.

[10] C.Chao-hung& C. Cheng-Der. HMM-based Part-of-Speech Tagging for Chinese Corpora. Proceeding of the Workshop on Very Large Corpora, Columbus, Ohio, 1993.

[11] M. Padró, L. Padró: Developing Competitive HMM PoS Taggers Using Small Training Corpora. 4th International Conference, EsTAL 2004, Alicante, Spain, 2004.