ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 2, February 2013

A Combined Approach to Part-of-Speech Tagging Using Features Extraction and Hidden Markov Model

Bhairab Sarma
Lecturer, Faculty of Science & Technology, The Icfai University Tripura, Kamalghat, Tripura-799210

Prajadhip Sinha
Asst. Professor, Department of Computer Science, Kohima Science College, Kohima, Nagaland

Dr. Bipul Shyam Purkayastha
Professor, Computer Science Department, Assam University, Silchar 788011

Abstract

Words are characterized by its features. In an inflectional language, category of a word can be express by its tense, aspect and modality (TAM). Extracting features from an inflected word, one can categorised it with proper morphology. Hence features extraction could be a technique of part-of-speech (POS) tagging for morphologically inflected languages. Again, many words could have same features with distinguish meaning in context. However contextual meaning could be recovered using Hidden Markov Model (HMM). In this paper we try to find out a common solution for part-of-speech tagging of English text using both approaches. Here we attempt to tag words with two perspectives: one is feature analysis where the morphological characteristics of the word are analyse and second is HMM to measure the maximum probability of tag based on contextual meaning with previous tag.

KEYWORDS: HMM, POS, contextual meaning, features extraction, TAM

1. Introduction

Part-of Speech Tagging is an activity to assign a proper tag (symbol) to each word from a written text or corpora. The primary objective of Part-of-Speech tagging (POST) is to word-category disambiguation. It is the process of marking up the words in a text (corpus) as corresponding to a particular part of speech, based on both its definition, as well as its context —i.e. relationship with adjacent and related words in a phrase, sentence, or paragraph[1]. The goal of POST is to identify words as nouns, verbs, adjectives, adverbs in a sentence according to the context as they appear to represent a meaning. Problem in here is for the words having multiple meaning[2] . There are many approaches to retrieve the sense of a word when they are playing a particular role in the context. [2] discussed an approach to express multiword expression where a tree technology is used for context retrieval. Hidden Markov Models is a common technique being used in POST since 1980. Scott and Harper (2008) used HMM to disambiguate parts of speech, during working to tag the Lancaster-Oslo-Bergen Corpus of British English in[3]. This involves counting cases such as from the Brown Corpus as in[4], and making a table of probabilities of certain sequences. For example, if there is an article 'the' in the sentence, perhaps the next word is a noun 40% of the time, an adjective 40%, and a number 20%. Retrieving these information's, a program can take a decision based on the maximum probability of the category. Rantaparikhi (1996) in [5] proposed his maximum entropy model where he claimed about 95% accuracy in part-of-speech tagging of structural language. With this entropy model, maximum probability of a tag could be determined without using HMM where comparatively more mathematical computations were required for higher accuracy level. Mark Jhonson (2007) in [6] pointed some reasons in favour of EM model comparative to HMM. Knowledge about the contextual meaning could be represented for the succeeding words in text. A second order HMM learn the probability up to two previous words in the sentence where as in higher order learn for triples or more. Higher in order of HMM used in POST, better is the accuracy result of tagging. Dynamic programming method is one alternative, developed by Steven DeRose and Ken Church (1987) with similar technique used in Viterbi algorithm. Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states – called the Viterbi path

– that results in a sequence of observed events, especially in the context of Markov information sources, and more generally, hidden Markov models. The forward algorithm is a closely related algorithm for computing the probability of a sequence of observed events. These algorithms belong to the realm of information theory.

Viterbi algorithm makes a number of assumptions.

- Two states of event sequences correspond to time: observed events and hidden events.
- There is a one-to-one mapping of these two sequences. An instance of an observed event needs to correspond to exactly one instance of a hidden event.
- A transition which computing the most likely hidden sequence up to a certain point *t* must depend only on the observed event at point *t*, and the most likely sequence at point $t - 1$.

These assumptions are all satisfied in a first-order hidden Markov model. But DeRose as in [3] used a table of pairs, while Church used a table of triples and an ingenious method of estimating the values for triples that were rare or nonexistent in the Brown Corpus. Both methods claimed accuracy over 95%.

2. Our approach

Taggers are broadly classified as: Rule-based, Stochastic and Hybrid (Eric Brill, 1992). A stochastic tagger falls under unsupervised category[7]. Unsupervised tagging techniques use an untagged corpus for their training data and produce the tagset by induction. Here patterns in word are observed, and derive part-of-speech categories themselves. For example, in most of the cases, "the", "a", and "an" occur in similar contexts, while "eat", "run" occurs in different ones. With sufficient iteration, similar classes of words emerge that are remarkably similar to those human linguists would expect; and the differences themselves sometimes suggest valuable new insights. In[8], Adam and Hal proposed a lexicon independent unsupervised part-of-speech tagger using HMM approach. It reveals that lexicon is not an essential component to build a tagger.

Our approach is a stochastic in nature perhaps falls in unsupervised category but not depend on HMM only. Brill (1992) in [7] demonstrate that the stochastic method is not only viable method for part of speech tagging. Here we combine two approaches; features extraction with morphological analysis and HMM. The first approach is used to tag closed word which are minimum in numbers and the second approach is used to smooth the probability of tagging for confused tags. For this purpose, we tag a text manually and extract the features of each unique word from the text depending on the assigned tags. We consider the relevance of previous tags as one of the feature of the word which reflects the application of Markov property. In this study, we collect the sample data from the first chapter of the book "Natural Language Processing: a Paninian Perspective". In [9], the authors Bharati et al. gives a brief introduction and overview of natural language processing including prospect and some key area of NLP development. They also describe some application area of NLP in this chapter. We tag manually 1409 root word based on morphological features. Depending on the tag frequency, we classify the entire tag into two classes as open and closed class. Closed class tags are taken special consideration in tagging. We used Penn Treebank tag set that consist of 45 tags. Out of all 45 tags, we exclude 10 punctuation marker tags and consider 35 tags for study. We follow the tagging guideline provided by Santorini of University of Pennsylvania, Philadelphia[1].

3. Observation

In this study, we first collect the word list by a special module called 'tokenizer' that tokenize individual token from the text file. Here, we tokenize the text with multiple successions. First white space is used to segregate the independent word, and then take care of about the ending character of each word. If the ending character is a special character (i.e. ., ;, :, ? etc), the word is again segregate in two tokens. If two consecutive words begin with capital character, we group them in a one token and assign a probable tag as 'NNP'. The frequency of each word is counted by a separate module called 'word-frequency-count'. This module is used to count the number of appearance of a particular word in the text. The system has been developed with PHP and MySql as front and back end respectively. We collect total 1409 word and

[1] Part-of-S peech Tagging Guidelines For The Penn Treebank Project (3rd Revision) MS-CIS-90-47 LINC LAB 178; Presented by Beatrice Santorini, Department of Computer and Information Science School of Engineering and Applied Science University of Pennsylvania Philadelphia, PA 19 104 July 1990 available at: http://repository.upenn.edu/cis reports/570

tagged manually according to Penn Treebank tagset and the frequency of tags are given in table 1.

Table 1: Tag Frequency

CC	62	JJS	3	VBG	28
CD	39	LS	4	PRP	7
DT	115	MD	7	PRP$	34
EX	8	NN	312	RB	21
FW	0	NNS	111	RBR	7
IN	160	NNP	17	RBS	2
JJ	206	NNPS	2	RP	0
JJR	4	PDT	6	SYM	0
TO	31	VB	92	VBD	59
VBN	28	VBP	3	VBZ	14
WDT	14	WP	1	WRB	7

Our approach does not follow HMM directly for context retrieval. In HMM, the meaning of a word W_i is depend on the meaning of its previous word sequence. However, in this approach it is difficult to predict the tag of the first word of the sentence. According to[10], in unsupervised method, no previous information is available in the system for tagging. Our approach is slightly different from this approach. We consider HMM finding tag transition probability for those words which have same features and confused in tagging them as per their appearance. Here we take the advantages form both HMM and features extraction method so that maximum possibility could be determined. For example, capitalized feature identified as noun, however each line is started with a capital letter. So, if the word is not a beginning word, and if it is not an abbreviation, the probability of being a noun is maximum. In next iteration, we take the transition matrix and find the transition probability for the tag. If the beginning word is a determiner for the next i.e. the previous tag is DT, then the possibility of the word to be noun is maximum. Hence the word would be tagged as a noun because:

1. The word is not a beginning word and it start with capital letter
2. The word is not an abbreviation because next character is not capital letter
3. The previous tag is a DT.

Next we identify the unique words that falls in same category. For example, in the third row of table 1, DT is given as 115 i.e. in the selected text, 115 times articles appeared. However in table two we show the number of unique word that falls in a particular category. For example in row two of table 2, DT is given three only. Meaning is that in our specimen text, although 115 of DT category word exist, only three separate word are exist in this category, and these are 'a', 'an', and 'the'. We found total 334 unique words in this table.

From these collections of words, we analyze each group of word morphologically and derive some features that hold in each word. These features are used to predict a probable tag for an unknown word. All features are enlisted in rows against each open category tags to form a features matrix. We show the matrix in table 5. Finally, we find the maximum probability of a tag by considering all features that hold in the specimen word and assign the tag to the word. For example, in case of tag NNS, the following features are observed:

1. If the word is ended with 's' or 'es'
2. The root word is 'noun' category
3. The previous tag is JJ
4. The word is start with a capital letter.

Table 2: List of Number of word in unique category

Tag category	No. of Words	Tag category	No. of words	Tag category	No. of Word	Tag category	No. of Words
CC	8	CD	40	EX	1	IN	6
DT	3	JJ	29	JJR	4	JJS	4
LS	8	MD	6	NN	39	NNS	27
NNP	17	NNPS	6	PDT	6	PRP	5
RBR	7	RB	17	PRP$	12	RBS	4
VB	24	VBD	32	VBG	22	VBN	17
VBP	2	VBZ	12	WDT	3	WRB	5
WP	2	WP$		SYM			

For a particular tag, if the number of word collection is less than 10% of total tags/word, the word is considered as rare word and that tags are not analysis for features extraction. Table 3 shows the sample size of unique word per tags. For example, in EX tags (in Table 2), we found only one word i.e. existential 'there'. Hence no features

are valid for EX tag except the word 'there'. Sometimes 'there' could be used as the beginning word but it is not classified as noun.

Table 3: Unique word per tag

Sample size	Unique word	Number of Tag	Average no. of W/T
1409	334	31	10.77

The following tags do not come under study due to small number (rare word) of word exists in the sample. In this case we simply tagged with specific word. For example, as stated previously, under EX tag we have only one word. So, the only feature is 'word' and consider as unique features, not applicable to other tags.

1. If the word is 'there' then the word tagged as 'EX'.

Few tags falls under this closed group is given below:

CC: Coordinating conjunction. Word group is "or", "and", "also", "but", "either", "neither", "nor". ,

DT: Determiner. If the word is either "a", "an", "the", "every", "any", "some", "each", "ether", "neither", "this", "that", "these", "those".

EX: Existential there –"there"

IN: Interjection group of word consist "by", "at", "for", "in", "of", "on", "under", "over".

MD: Modal verb, a type of auxiliary verb. Modal verbs do not take the inflection *-s* or *-es* in the third person singular, unlike other verbs. Includes "shall", "should", "will", "would", "can", "could", "may", "might", "mote", "must".

WDT: Words include "which", "that".

WP: Word include "what", "why", "who", "whom".

WRB: In this group the word includes "how", "however", "whereas", "whether", "where".

3.1 The Features Matrix

We consider the following tags for analysis. These tags are frequently used in text and shows ambiguity in tagging. Words found here are very confusing in their sense and considered as open class. Following table 4: shows some confused tags with their frequencies.

Table 4: Confusion tags with frequency

NNP	16	VB	24	CD	40	JJ	23	RB	17
VBD	32	VBG	22	NN	30	PRP$	12	VBZ	12
NNS	19	VBN	7						

From this list, we observed some peculiar features that exist in these words and find the relevancy of their category by a probabilistic factor (Fc). We measure this factor in terms of percentage using the formula as:

$$Fc = \frac{\text{number of word that hold that features}}{|\text{number of unique word in that category}|} x100\%$$

For instance in 'NN' groups, there are 40 unique words and out of 40, fifteen words are ended with "ive", then the rank of that features to be grouped under 'NN' is 15/40*100=37.5%. For each rows, we compute Fc value for all tags and take the transition values from transition matrix. Finally, we compute the average of both and find the maximum value of that average. During tagging we take a word from the text, check it whether it falls under rare group and the if not, we analyse the features matrix and for each tag and sum with the rank given by previous tag probability from transition matrix (HMM) and assign that particular tag which give the maximum average. For example: if given a word ended with "ogy" (features-last three characters), we move to the second row of the feature matrix and read each value for each category and look in transition matrix for that category and compute the average. The features matrix is given in table 5.

3.2 The Transition Matrix

The transition matrix is a square matrix computed from confusion tags of the corpus. Here we consider fifteen confusion tags and compute the transition probabilities as given below:

Table 5: The Features-rank Matrix

Features/Tags		NN	NNS	JJ	NNP	VB	VBG	VBN	VBZ	VBD	RB	PRP$
Begins with cap		23.08	11.11	17.24	100.00	4.17	4.55	0.00	0.00	0.00	0.00	8.33
Beginning word of a line		20.51	11.11	0.00	17.65	8.33	9.09	0.00	0.00	0.00	0.00	0.00
All cap		5.13	0.00	0.00	47.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Suffixes	**s**	2.56	74.07	0.00	0.00	0.00	4.55	5.88	50.00	3.13	11.76	16.67
	es	0.00	11.11	0.00	0.00	0.00	0.00	0.00	33.33	0.00	0.00	0.00
	al	2.56	0.00	13.79	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ed	0.00	0.00	10.34	0.00	4.17	0.00	0.00	8.33	50.00	0.00	0.00
	ly	0.00	0.00	6.90	0.00	0.00	0.00	0.00	0.00	0.00	35.29	0.00
	tic	0.00	0.00	6.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ess	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ses	0.00	7.41	6.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ion	10.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ive	2.56	0.00	3.45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ogy	2.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ism	2.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ing	0.00	0.00	0.00	0.00	4.17	63.64	52.94	0.00	0.00	0.00	0.00
Prefixes	**im**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	de	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	un	0.00	0.00	0.00	0.00	0.00	0.00	5.88	8.33	6.25	0.00	0.00
	in	5.13	10.53	6.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Transition probability is computed as:

$$p(t_2 \mid t_1) = \frac{C(t_1 \& t_2)}{C(t_1)}$$

For example determining the probability of noun following by a determiner is

$$p(noun \mid \det) = \frac{c(\det \& noun)}{c(\det)}$$

That is number of determiner and noun occurred together divided by the number of times that a determiner occurred. In table 6 we show the transition probability matrix of 15 confusion tags-

4. Findings

We test our model in various text corpora by inputting simple text file. Words which are not fall in our consideration, we don't tag them in confusion and we simply kept as untagged word. The accuracy matrix of our experiment is given in table 7. For instance we input a text consists of 627 words and it tag accordingly and found the following result. Out of 627 words, 40 are unable to tag and remain as untagged, which include those words that do not hold any features or may be foreign word. Some punctuation marks are also included in this category. Large numbers of words that are assigning wrongly because of less number of observation samples were taken for features consideration. For tags 'CC', 'IN', 'DT', 'RBR', 'TO', 'EX' etc., we achieve almost 100% accuracy due to their direct word consideration and we exclude these from our analysis. Considering only the confusion words, we achieve almost 70% accuracy. A very good figure of accuracy level is found in tags 'JJ', 'VBN', 'VBG' and 'VBZ' due to their unique features. The overall accuracy will increase up to 80% including rare word category.

Table 6: Transition Probability Matrix

	NN	NNP	NNS	CC	CD	DT	IN	JJ	PRP$	RB	VB	VBG	VBN	VBD	VBZ
NN	16.35	1.6	8.33	6.41	4.49	3.21	21.15	9.29	1.6	0.64	8.01	3.21	0.96	1.6	2.24
NNP	11.76	11.76	0	5.88	35.29	0	23.53	0	0	0	5.88	0	0	5.88	0
NNS	11.71	0.9	6.31	9.01	0	1.8	21.62	15.32	1.8	0.9	9.91	4.5	1.8	4.5	0
CC	20.97	3.23	4.84	0	0	12.9	4.84	17.74	4.84	1.61	4.84	1.61	1.61	8.06	3.23
CD	7.69	0	12.82	7.69	7.69	5.13	5.13	12.82	5.13	5.13	2.56	0	0	2.56	2.56
DT	58.26	0	4.35	0	1.74	0	0	22.61	0	6.96	0	0.87	1.74	0.87	0
IN	62.26	0.94	25.47	6.6	2.83	16.04	21.7	29.25	2.83	0.94	2.83	2.83	4.72	4.72	0
JJ	32.04	0.49	13.11	3.4	1.46	8.25	11.17	15.05	1.46	0.49	1.46	1.46	2.43	2.43	0
PRP$	26.47	0	5.88	5.88	0	0	5.88	8.82	0	2.94	20.59	0	0	5.88	8.82
RB	4.76	0	19.05	4.76	14.29	19.05	0	4.76	4.76	9.52	0	0	0	14.29	0
VB	7.61	1.09	2.17	4.35	1.09	8.7	9.78	13.04	5.43	1.09	2.17	4.35	11.96	17.39	0
VBG	14.29	3.57	39.29	0	0	3.57	0	17.86	0	0	7.14	0	0	3.57	0
VBN	10.71	0	10.71	0	0	3.57	10.71	25	7.14	3.57	3.57	0	3.57	14.29	0
VBD	8.47	0	0	10.17	1.69	10.17	25.42	23.73	0	0	0	3.39	0	1.69	0
VBZ	7.14	0	0	0	7.14	28.57	7.14	7.14	0	0	14.29	0	0	7.14	0

Some confused words are tagged as rare category for which the number of incorrect tags being increased specifically in 'VB', 'NN' tag, because these words directly match with the rare category.

Table 7: Accuracy level of confused tags

Tag	Number of word	Correctly tag	Wrongly tag	Accuracy %
NN	127	87	40	68.5
NNP	116	90	26	77.59
JJ	54	44	10	81.48
PRP$	46	34	12	73.91
VB	98	66	32	67.35
VBN	47	37	10	78.72
VBG	32	26	6	81.25
VBD	35	29	6	82.86
VBZ	24	21	3	87.5
Untagged	48			
Total	627	434	145	69.21

5. Conclusion

In this paper, we try to find a weight of a tag in two perspectives: one by features analysis and the other by using HMM. Feature extraction method could be enhanced using constraint grammar satisfaction method. Constraint Grammar (CG) as proposed by Fred Karlsson (in 1990) is a methodological paradigm for Natural language processing. In this approach linguist-written, context dependent rules are compiled into a grammar that assigns grammatical tags ("readings") to words or other tokens in running text. In future we suppose to imply this approach to achieve maximum weightage with higher order HMM. This method could be enhance to multiple perspective model which could include rule based, constraint satisfaction, HMM and features extraction method.

References

[1] Daniel Jurafsky & James H. Martin, WORD CLASSES AND PART OF SPEECH TAGGING, Speech and Language Processing: *An introduction to natural language processing, computational linguistics, and speech recognition.* 2005

[2] Spence Green, Marie-Catherine de Marneffe, John Bauer, and Christopher D. Manning, Multiword expression identification with tree substitution grammars: a parsing tour de force with French**, *in*** *EMNLP '11 Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ACL 2011

[3] Scott M. Thede and Mary P. Harper, A Second-Order Hidden Markov Model for Part-of-Speech Tagging, *In Proceedings of the 37th Annual Meeting of the ACL*

[4] Julia Hockenmaier , Building a (statistical) POS tagger: HMM POS-tagging with Hidden Markov Models at 3324 Siebel Center

[5] Adwait Ratnaparkhi, A Maximum Entropy Model for Part-Of-Speech Tagging, *In Proceedings of the Empirical Methods in Natural Language Processing* (1996), pp. 133-142.

[6] Mark Johnson, Why doesn't EM find good HMM POS-taggers? Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 296–305, Prague, June 2007. c 2007 ACL

[7] Eric Brill. 1992. A simple rule-based part of speech tagger. *In Proceedings of the third conference on Applied natural language processing* (ANLC '92).

[8] Adam R. Teichert , Hal Daume III, Unsupervised Part of Speech Tagging without a lexicon, School of Computing, university of Utah, 2009

[9] Akshar Bharati, Vineet Chaitanya Rajeev Sagal, Natural Language Processing, A Paninian Perspective, Prentice-Hall of India, New Delhi

[10] Doug Cutting and Julian Kupiec Jan Pedersen and Penelope Sibun, A Practical Part-of-Speech Tagger, *In Proceeding of the third conference on applied Natural language Procesing*, Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304, USA (1992)