VOL. 11, NO. 13, JULY 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

EXPERIMENTAL ANALYSIS OF MALAYALAM POS TAGGER USING EPIC FRAMEWORK IN SCALA

Sachin Kumar S., M. Anand Kumar and K. P. Soman
Centre for Excellence in Computational Engineering and Networking, Amrita Vishwa Vidyapeetham, Coimbatore, India
E-Mail: sachinnme@gmail.com

ABSTRACT

In Natural Language Processing (NLP), one of the well-studiedproblems under constant exploration is part-of-speech tagging or POS tagging or grammatical tagging. The task is to assign labels or syntactic categories such as noun, verb, adjective, adverb, preposition etc. to the words in a sentence or in an un-annotated corpus. This paper presents a simple machine learning based experimental study for POS tagging using a new structured prediction framework known as EPIC, developed in scale programming language. This paper is first of its kind to perform POS tagging in Indian Language using EPIC framework. In this framework, the corpus contains labelled Malayalam sentences in domains like *health*, *tourism* and *general* (news, stories). The EPIC framework uses conditional random field (CRF) for building tagged models. The framework provides several parameters to adjust and arrive at improved accuracy and thereby a better POS tagger model. The overall accuracy were calculated separately for each domains and obtained a maximum accuracy of 85.48%, 85.39%, and 87.35% for small tagged data in *health*, *tourism* and *general* domain.

Keywords: parts-of-speech tagging (POS), conditional random field (CRF), AMRITA tag set, EPIC, Malayalam language.

1. INTRODUCTION

The part-of-speech (POS) tagging is a well-known problem under constant research in language processing [1]. A POS tagger is an essential tool for parsing, information retrieval, word sense disambiguation, correct lemmatization etc. POS tagging is the process by which the words in the sentence are assigned with tags that shows its syntactic category depending on the context. Or a method by which words in a language are categorized depending on the morphological and syntactic features. The common categories for tag are noun, verb, adverb, adjective, conjunction etc. POS tagging plays an important role in applications like machine translation, language modeling, word sense disambiguation, Question and Answer analysis, dialogue tagging, social media data tagging, information retrieval etc. For example, the following Malayalam word ഇരുൾ denotes a verb and noun as it has two meaning - ഇരുട്ടുക, കറുക്കുക and ഇരുപ്പൂൾ, ഒരുവൃക്ഷം. Therefore, the task of the POS is to disambiguate and correctly identify the grammatical category.

In the Indian language scenario, POS taggers were developed for Dravidian languages (Kannada, Malayalam, Tamil and Telugu), Hindi, Punjabi, Odia, Marathi and Bengali. Each language have their own tag set prepared by different organization or research groups and it will contain main tags and sub tags which refers its morpho-syntactic features [2]-[15]. The Bureau of Indian Standards (BIS) POS tag set for Indian languages aims to ensure a common language tag set for Indian languages. It was prepared by POS tag standardization Committee, Department of Information Technology, New Delhi.

Several methods are applied for POS tagging task. In [16], [17], [18] discusses hidden markov model based POS tagging, memory based learning [19], maximum entropy modeling [20], transformation based learning [21], decision trees [22], [23], support vector machines [24], [13], rule based approach [25], using disambiguation rule [26], [27], hybrid approaches are also been made using stochastic method and rules [28]. Indian languages are morphologically rich and this posses major challenge in disambiguating words thereby the number of tags required will be more to deal with ambiguities. The morphological richness of the language creates difficulty to prepare complex rules for POS tagging. The machine learning approaches uses the linguistically motivated data associated with each language. Due to high inflective nature of the Indian languages, the method/techniques used for one language may not be useful for the other. Several articles for POS tagging the morphologically rich language were proposed in which the stochastic methods and specific hand crafted rules with the help of linguist were developed [29], [30], [31], [32]. This approach raises the requirement of an expert linguist opinion to create accurate rules and large corpus for stochastic methods to be effective. Several approaches related to POS tagging in Malayalam language is also carried out [13], [45]. This paper presents a POS tagger for Malayalam language using EPIC framework in scale language. In this, the POS tagging task is defined as a sequence labeling problem. This is a first attempt to explore the EPIC framework for POS tagging in Indian languages.

This paper is organized as follows. Section 'Tagset' gives an overview about AMRITA tag set. Section 'Condition Random Fields' gives a brief introduction about condition random fields. Section ' EPIC framework ' gives an overview about the EPIC framework. In section 'Experimental Result', the experiments and the obtained results are discussed.

1.1 Tag set

A tag set represents the tag categories that can be used to tag each word based on the context. Several researchers in Indian language uses different tag set such as AUKBC, Vasuranganathan tag set, CIIL Tag set,

TDILetc). The tag set posses the following issues - (1) the grammatical categories and grammatical features are considered for each word. This will increase the complexity when words with inflections are encountered.(2) Increase in the number of tags in the tag set increases the complexity in the POS tagging also. Therefore, without compromising the efficiency in tagging, AMRITA tag set was developed by following the guidelines mentioned in AnnCorra [39] and EAGLES [41]. This paper followed AMRITA tag set for tagging each word in the corpus [38] in *general* domain. The tag set contains 29 tags. The tag considers the category of words and not the grammatical features and the inflections. Table 1 and 2 shows the AMRITA tag set. The AMRITA tag set contains 5 tags for nouns, 3 tags for punctuations, 7 tags for verb, 1 tag for pronoun, adverb, adjective, conjunction, postposition, determiners, echo, comma, emphasis etc.

Table-1. AMRITA tag set.

S. No.	Tag	Description
1	<NN>	NOUN
2	<NNC>	COMPOUND NOUN
3	<NNP>	PROPER NOUN
4	<NNPC>	COMPOUND PROPER NOUN
5	<ORD>	ORDINALS
6	<CRD>	CARDINALS
7	<PRP>	PRONOUN
8	<ADJ>	ADJECTIVE
9	<ADV>	ADVERB
10	<VNAJ>	VERB NON FINITE ADJECTIVE
11	<VNAV>	VERB NON FINITE ADVERB
12	<VBG>	VERBAL GERUND
13	<VF>	VERB FINITE
14	<VAX>	VERB AUXIALLARY
15	<VINT>	VERB INFINITE
16	<CNJ>	CONJUNCTION
17	<CVB>	CONDITIONAL VERB
18	<QW>	QUESTION WORD
19	<COM>	COMPLEMENTIZER
20	<NNQ>	QUANTITY NOUN
21	<PPO>	POSTPOSITIONS
22	<DET>	DETERMINERS
23	<INT>	INTENSIFIER
24	<ECH>	ECHO WORDS
25	<EMP>	EMPHASIS
26	<COMM>	COMMA
27	<DOT>	DOT
28	<QM>	QUESTION MARKS
29	<RDW>	REDUPLICATION WORDS

Examples of words tagged using AMRITA tag set is
അവ<PRP>സാധാരണ<ADJ>യാത്രകളായി<NN>
മാറിക്കഴിഞ്ഞു<VNAV>. <DOT>.

1.2 Conditional random fields

Conditional Random Field (CRF) is a statistical modeling method used for machine learning and pattern recognition applications. This graphical model can suitably represent the dependency structure or to encode the relationship existing in the sequential data like in natural language text. Traditionally modeling was done by considers the joint density probability distribution, p(y,x), where 'y' denotes the set of features or attributes related to the entity required to be predicted and 'x' is the input variables which represents the observed knowledge. In this stochastic approach a joint probability value is assigned to the pair of observation sequence and its label which will be the maximum joint likelihood value. This modeling will fail when considering the relational data as it won't consider the dependency relation among the data. As a solution to this classification problem, became the motivation for the conditional model representation $p(y|x)$ of the pair of observation sequence and labels. In a conditional model, for a given observation sequence, the model specifies the probabilities possible to the label sequences [33], [34].

Let X denotes a random variable for observation sequences to be labelled, Y denotes a random variable on labelled sequences. All Y_i of Y belongs to a finite class $\mathcal{Y}$. For example, X represents the natural language sentences and Y represents the set of all possible POS tags. Using the two random variables X and Y, a conditional model, $p(Y|X)$, from paired observation and labels are constructed. A graphical model is a probabilistic model in which the dependency relation between the random variables is denoted by a graph. CRF is an undirected graphical model of the form

$$p(x,y) = \frac{1}{Z}\prod_{A}\Psi_A(x_A, y_A) \quad (1)$$

$$Z = \sum_{x,y}\prod_{A}\Psi_A(x_A, y_A) \quad (2)$$

Where $\Psi_A : \nu^n \to \Re^+$ denotes the local function or compatibility function. Equation (1) says that for a choice of factors $F = \{\Psi_A\}$, the probability distribution for many random variables can represented as the product of local functions. And each local function is depending on smaller number of random variables. The quantity Z is a normalization factor which keeps the probability distribution to 1 [34].

2. EPIC FRAMEWORK

EPIC is a scale based statistical parser developed by David Hall [35], [36], [37]. It can be used for building high performance structured prediction models for parsers, POS taggers, named entity recognition, segmenters etc. EPIC can be used as a command-line tool or programmatically to train our own models. Three kinds of models such as parsers, segmenters, and sequence labelers could be developed using EPIC. Sequence labeling is a task where the objects in sequence need to be labelled. For example words in a sentence. Here, the POS tagging is defined as a sequence labeling problem. In POS tagging operation each of the words are tagged or labelled under grammatical category as noun, verbs, adverb, adjectives etc. Segmentation is usually adopted as a pre-processing task. It breaks the sentences into different sequences or chunks. EPIC also supports training of different discriminative parsers. It provides few options to fine tune the training model. It provides 4 different base models such as LatentModelFactory which uses latent annotations (eg. Berkeley parser), LexModelFactory which uses lexical annotations (similar to Collins parser), StructModelFactory which uses structural annotations (similar to Standford parser), SpanModelFactory which uses span features. Currently EPIC provides pre-trained models for parser (for English, Basque, French, German, Hungarian, Korean, Polish, Swedish), POS tagger (for English, Basque, French, German, Hungarian, Polish, Swedish), and named entity recognition (English). EPIC also provide the facility to build or prepare model for own language. It utilizes SBT (Simple Build Tool) to compile and run or to build it. This process adds all the dependencies required. This is the first attempt which explores the use of EPIC for an Indian language. The SBT build will ensure that all the dependencies involved in building the package (.jar file) is handled. By using these models, programmatically we can train for the language specific to the model. EPIC also provides the option to programmatically develop the model for languages that doesn't have pre-trained models. These training are controlled by several options to provide optimized training. EPIC develops the model using linear chain conditional random fields. To bring the accurate model, it provides several tuning parameters and the choice of selecting the features like word features, span features (feature based on the entire input span), and split span features (features for begin, split and end) [33], [37], [40].

3. EXPERIMENTAL RESULT

The POS tagging model was collected on data from three domains such as health, tourism and general (collected from web). The data for *health* and *tourism* domain were obtained from [42] and contains 25000 annotated sentences. The sentences in *general* domain were manually tagged using AMRITA tag set discussed in section II and contains 13000 annotated sentences. The EPIC framework was run on 8Gb RAM machine using Eclipse IDE. Since 'Out of Memory' error was occurring for large data corpus, the size of the corpus was limited for this experiment and the details are iven in Table-1. Based on this observation, the corpus used for experiment contains 4138 annotated sentence in *health* domain, 4212 annotated sentence in *tourism* domain and 4799 sentences in *general* domain.

Table-2. Details of corpus.

Domain	T. Sentence	T. Words	Avg. words/sent
Health	4138	50011	12
General	4799	50968	10
Tourism	4212	50010	11

The first column in Table-1 denotes the corpus domain, second column denotes the total sentences in each corpus, third column denotes the total words in each corpus, and fourth column denotes average words/sentence. The prepared data was saved in '.csv' format with words in Malayalam language in first column containing and the second column contains the corresponding tag. The tagger model was tested using 10% of the total data corresponding to each domain and the remaining data were used for training. To train the model, simple features like unigram of words, unigram for word class, bigram of words, bigram of word class were taken. This shows the possibility of improving the accuracy by incorporating methods such as suffix, prefix, language specific features etc. The EPIC framework provides several parameters such as regularization, batch size, maximum iteration, useL1, use Stochastic to arrive at better accuracy. This paper presents the result by calculating accuracy value by varying the parameter values. The parameters useL1 and use Stochastic takes boolean values. The framework uses iterative algorithms such as adaptive gradient descent methods, Limited-memory BFGS (L-BFGS or LM-BFGS) [43] and Orthant-wise limited-memory quasi-Newton (OWL-QN) [44]. The parameter for use Stochastic and useL1 provides four different combinations to select the iterative algorithm with L_1 or L_2 regularization. The *use Stochastic* parameter if true, selects Stochastic Gradient Descent algorithm. If false, then it selects LBFGS or OWLQN. This combination is tabulated in Table-2. The *useL1* parameter if false, selects L_2 regularization and if true, it selects L_1 regularization. The parameter *batch size* will only be useful when the stochastic gradient descent methods are used. In Table-1, letter 'F', 'T' denotes 'False' and 'True'.

Table-3. Algorithm selection.

use Stochastic	useL1	Iterative Alg
F	F	LBFGS+L2
F	T	OWLQN+L1
T	F	SGD+L2
T	T	SGD+L1

In general, for 3 domains, it was observed that as the *batch size* increases, time taken for computation slightly increases. The *batch size* is varied as 100, 200,..1000. The *regularization* parameter is varied as 5, 4, 3, 2, 1, 0.5, 0.25, 0.05. It is a constant value. Also the *iteration* parameter is varied as 25, 50, 75...200. The overall accuracy of POS tagging is calculated by changing each parameter value iteratively. From the pool of accuracy values thus obtained, the highest accuracy gives the suitable values for each parameter that build the trained model. Execution of each parameter configuration consumes lot of time. The parameters *use Stochastic* and *useL1* gives four combinations with boolean values. For each such combinations, the parameter *iteration* is varied as 25, 50, 75...200. Then for each iteration, batch size is fixed and regularizations are varied to find the accuracy. This can be understood from Table-3. In Table-3, it shows the accuracy calculated for use Stochastic -false', useL1 -true', and for iteration-150. It can be observed that the for each batch size and regularization, accuracy is calculated. In this parameter configuration, accuracy values vary from lowest 66.96% to highest 82.23%. Since eight iterations values are used, eight such tables like Table-2 was prepared for each combination of *use Stochastic* and *useL1*boolean values. Therefore, for a single POS tagged data domain, there will be a total of 32 such Table like Table-2.

Table-4. Accuracy obtained by varying batch size and regularization.

B/R	5	4	3	2	1	0.5	0.25	0.05
100	73.85	71.97	78.41	75.55	76.37	79.91	80.94	79.70
200	69.33	76.45	68.50	77.85	79.54	77.34	79.58	81.60
300	72.46	77.64	76.68	75.55	79.30	80.98	80.11	81.12
400	74.76	72.27	73.02	74.21	80.45	76.62	81.14	81.42
500	68.26	70.78	79.76	78.31	73.69	75.89	80.39	79.85
600	75.50	74.25	78.84	77.62	78.82	80.41	78.53	79.79
700	66.96	77.81	75.87	80.57	73.06	78.80	75.67	79.81
800	71.69	71.37	74.62	79.00	80.57	77.32	80.19	82.23
900	72.60	74.76	76.74	80.09	75.79	79.18	81.00	81.88
1000	77.18	71.95	76.57	80.57	79.56	80.35	78.37	80.13

In order to find the highest accuracy, from each 32 accuracy tables thus obtained, maximum accuracy value from each table is noted and its shown in Tables 4, 5 and 6. The Tables 4, 5, and 6 represents the maximum accuracy corresponding to each boolean parameter configuration and iteration. Figures 1, 2, and 3 shows the corresponding plots of variation in accuracy wart iteration. The plot in green colour denotes 'FF' combination in *use Stochastic* and *useL1* parameters. The colours blue, red and violet represents, 'TF', 'TT', and 'FT' combinations.

Table-5. Health domain.

Iter	TF	TT	FF	FT
25	84.99	82.98	85.48	83.86
50	84.85	82.45	85.24	82.33
75	84.87	83.14	85.32	82.46
100	84.83	83.22	85.22	82.96
125	84.73	82.39	85.28	82.21
150	84.81	82.94	85.28	82.23
175	84.78	82.29	85.22	82.49
200	84.87	83.22	85.24	82.96

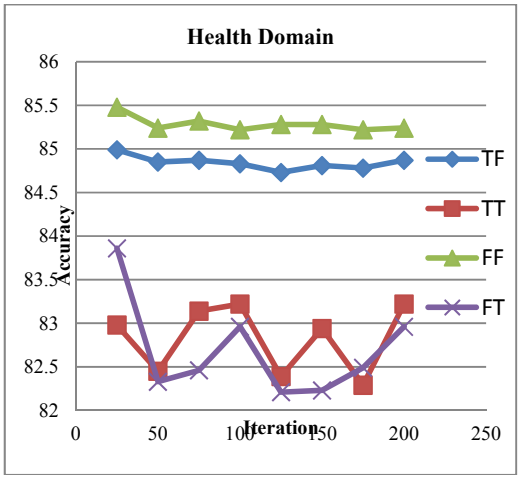

Figure-1. Plot of maximum accuracies from POS tagged data in health domain.

Table-6. General domain.

Iter	TF	TT	FF	FT
25	85.61	85.41	83.67	85.10
50	86.91	85.55	87.35	84.99
75	86.85	85.31	87.31	85.18
100	86.83	85.39	87.21	85.71
125	86.80	85.74	87.21	85.08
150	86.83	86.02	87.19	85.44
175	86.87	85.86	87.19	85.08
200	86.83	85.27	87.24	85.42

From Figure-1, it can be observed that for the tagged data in health domain, the combination of LBFGS iterative algorithm and L_2 regularization gives better accuracy. The overall accuracy of 85.48% was obtained for tagged data in *health* domain.

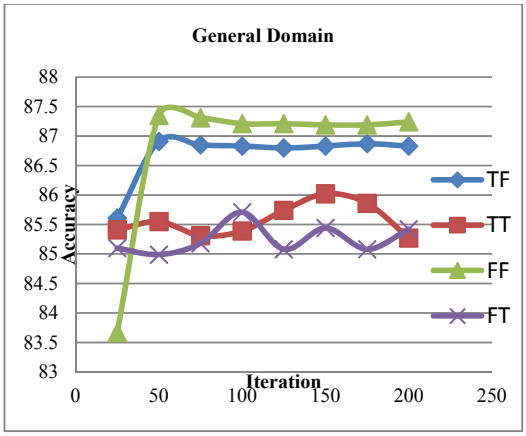

Figure-2. Plot of maximum accuracies from POS tagged data in General domain.

Table-7. Tourism domain.

Iter	TF	TT	FF	FT
25	82.64	84.49	83.62	83.04
50	84.75	85.00	85.37	82.23
75	84.85	84.81	85.36	83.37
100	84.83	84.81	85.39	83.30
125	84.81	84.81	85.37	82.70
150	84.89	84.85	85.32	82.94
175	84.81	84.92	85.34	83.02
200	84.75	84.78	85.34	82.79

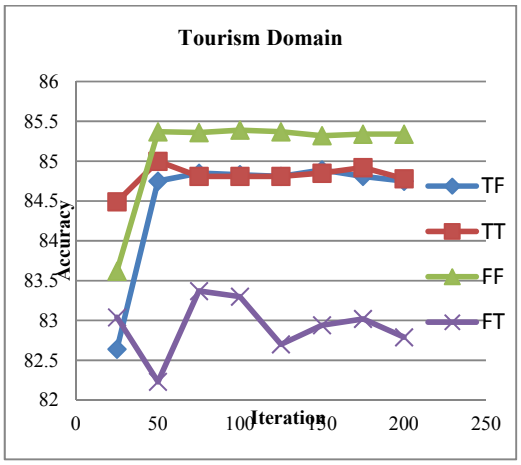

Figure-3. Plot of maximum accuracies from POS tagged data in Tourism domain.

Figures 2 and 3 gives an overview of accuracy variation for different iterative algorithms and regularization as shown in Table 1 for tagged data in *General* and *Tourism* domain. It can be observed that the highest accuracies obtained for both domain are 87.35% and 85.39%. From Figure 1, 2 and it can be observed that LBFGS with L_2 regularization gives better accuracy.

4. CONCLUSION AND FUTURE WORK

This paper described about the implementation of part-of-speech tagger using EPIC framework. The data consists of annotated sentences form three domains such as *health*, *general* and *tourism*. The training data in each domain was modeled using conditional random fields so that the relation from neighbour words can be utilized completely. Simple features like unigrams and bigrams were utilized and obtained an overall accuracy of 85.48%, 87.35%, and 85.39% for tagged data in *health*, *general* and *tourism* domain. The framework provides certain parameters which helps in improving the accuracies. The future work is to solve the memory issue and build a model for atleast 25000 sentences. EPIC framework gives the possibility of adding suitable language specific features, stems, suffix, prefix etc., will perform another experimental study based on new features and large tagged corpus.

REFERENCES

[1] Jurafsky D and Marting J H. 2002. Speech and Language Processing an Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Pearson Education Series.

[2] Dinesh Kumar and Gurpreet Singh Josan. 2010. Part of Speech Taggers for Morphologically Rich Indian Languages: A Survey. International Journal of Computer Applications (0975-8887). 6(5).

[3] Manish Shrivastava and Pushpak Bhattacharyya. 2008. Hindi POS Tagger Using Naive Stemming: Harnessing Morphological Information without Extensive Linguistic Knowledge. Department of Computer Science and Engineering. Indian Institute of Technology, Bombay. Proceeding of the ICON 2008.

[4] Nidhi Mishra Amit Mishra. 2011. Part of Speech Tagging for Hindi Corpus. International Conference on Communication Systems and Network Technologies.

[5] PradiptaRanjan Ray, Harish V. SudeshnaSarkar and AnupamBasu. Part of Speech Tagging and Local Word Grouping Techniques for Natural Language Parsing in Hindi. Department of Computer Science and Engineering. Indian Institute of Technology. Kharagpur. INDIA 721302 www.mla.iitkgp.ernet.in/papers/hindipostagging.pdf.

[6] DebasriChakrabarti. 2011. Layered Parts of Speech Tagging for Ban gla. Language in India www.languageinindia.com, May 2011. Special Volume:Problems of Parsing in Indian Languages.

[7] Vijayalaxmi .F. Patil. 2010. Designing POS Tagset for Kannada. Linguistic Data Consortium for Indian Languages (LDC-IL). Organized by Central Institute of Indian Languages. Department of Higher Education Ministry of Human Resource Development. Government of India. March.

[8] Hammad Ali. 2010. An Unsupervised Parts-of-Speech Tagger for the Bangla language. Department of Computer Science. University of British Columbia.

[9] S. Rajendran. 2006. Parsing in Tamil. LANGUAGE IN INDIA. www.languageinindia.com.6: 8.

[10] M. Selvam, A.M. Natarajan. 2009. Improvement of Rule Based Morphological Analysis and POS Tagging in Tamil Language via Projection and Induction Techniques. International Journal of Computers. 3(4).

[11] Dhanalakshmi V1, Anand Kumar1, Shivapratap G1, Soman KP1 and Rajendran S. 2009. Tamil POS Tagging using Linear Programming. International Journal of Recent Trends in Engineering. 1(2).

[12] Dhanalakshmi V, Anandkumar M, Rajendran S, Soman K P. POS Tagger and Chunker for Tamil Language.

[13] Antony P J, Santhanu P Mohan and Soman K P. 2010. SVM Based Parts Speech Tagger for Malayalam. International Conference on-Recent Trends in Information. Telecommunication and Computing (ITC).

[14] A Part of Speech Tagger for Indian Languages (POS tagger). 2007. Tagset developed at IIIT-Hyderabad after consultations with several institutions through two workshops. shiva.iiit.ac.in/SPSAL2007/iiit_tagset_guidelines.pdf.

[15] G.M. Ravi Sastry, Sourish Chaudhuri and P. Nagender Reddy. 2007. An HMM based Part -Of-Speech tagger and statistical chunker for 3 Indian languages. Shallow parsing for South Asian Language Workshop. International joint Conferences on Artificial intelligence. www.cs.cmu.edu/~schaudhu/publications.html.

[16] T. Brants. 2000. TnT - a statistical part-of-speech tagger. In: Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000.

[17] K. W. Church. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In: Proceedings of the Second Conference on Applied Natural Language Processing. pp. 136-143.

[18] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. 1992. A practical part-of-speech tagger. In: Proceedings of the Third Conference on Applied Natural Language Processing. pp. 133-140.

[19] W. Daelemans, J. Zavrel, P. Berck, and S. Gillis. 1996. Mbt: A memory-based part of speech tagger-generator. In:Ejerhed, E. and Dagan, I. (editors). Proceedings of the Fourth Workshop on Very Large Corpora. pp. 14-27.

[20] A. Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, University of Pennsylvania.

[21] E. Brill. 1992. A simple rule-based part of speech tagger. In Proceedings of the Third Conference on Applied Natural Language Processing, Trento, Italy.

[22] E. Black, F. Jelinek, J. Lafferty, R. Mercer, and S. Roukos. 1992. Decision tree models applied to the labeling of text with parts of speech. In: Proceedings of the DARPA Speech and Natural Language Workshop. Arden House. NY.

[23] L. M`arquez, and L. Padr´o. 1997. A flexible POS tagger using an automatically acquired language model. In: Proceedings of the 35th Annual Meeting of the ACL. pp. 238-245.

[24] J. Gim´enez, and L. M`arquez. 2004. SVMTool: A general POS tagger generator based on support vector machines. In: Proceedings of the IV International Conference on Language Resources and Evaluation (LREC'04), pp. 43-46.

[25] B. Eric. 1992. A simple rule based part of speech tagger. In Proceedings. Third Conference on Applied Natural Language Processing. ACL. http://www.aclweb.org/anthology/A92-1021 (visited on 05/7/2015).

[26] P.Tapanainen and A. Voutilainen. 1994. Tagging accurately - don't guess if you know. In Proceedings of the 4th Conference on Applied Natural Language Processing. pp. 47-52.

[27] B. Greene and G. Rubin. 1971. Automatic grammatical tagging of English. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island.

[28] R. Garside and N. Smith. 1997. A hybrid grammatical tagger: Claws4. In R. Garside, G. Leech, A. McEnery (eds.) Corpus annotation: Linguistic information from computer text corpora. pp. 102-121.

[29] J. Hajic, P. Krbec, P. Kveton, K. Oliva and V. Petkevic. 2001. A case study in czech tagging. In Proceedings of the 39th Annual Meeting of the ACL.

[30] Y. Tlili-Guiassa. 2006. Hybrid method for tagging arabic text. Journal of Computer Science. 2(3): 245-248.

[31] K. Uchimoto, S. Sekine, and H. Isahara. 2001. The unknown word problem: a morphological analysis of Japanese using maximum entropy aided by a dictionary. In: Proceedings of the Conference on EMNLP.

[32] K. Oflazer and I. Kuruoz. 1994. Tagging and morphological disambiguation of Turkish text. In: Proceedings of the 4 ACL Conference on Applied Natural Language Processing Conference.

[33] An introduction to conditional random fields for relational learning.

[34] http://people.cs.umass.edu/~mccallum/papers/crf-tutorial.pdf (visited on 01/7/2015).

[35] J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proc. 18th International Conf. on Machine Learning.

[36] http://dlwh.org/ (visited on 05/7/2015).

[37] https://github.com/dlwh/epic (visited on 05/7/2015).

[38] D. Hall, G. Durrett, and D. Klein. 2014. Less grammar, more features. in ACL.

[39] https://www.amrita.edu/center/computational-engineering-andnetworking/research/computational (visited on 05/7/2015).

[40] A. Bharati, D. M. Sharma, L. Bai, and R. Sangal. 2006. AnnCorra: Annotating Corpora Guidelines for POS and Chunk Annotation for Indian Languages. Language Technologies Research Centre, IIIT, Hyderabad.

[41] David Hall and Dan Klein. 2012 Training Factored PCFGs with Expectation Propagation. In EMNLP.

[42] www.corpora.dslo.unibo.it/TCORIS/EAGLES-like_POSTagset.pdf (visited on 05/7/2015).

[43] http://tdil-dc.in/ (visited on 05/7/2015).

[44] Dong C. Liu, Jorge Nocedal. 1989. On the Limited memory BFGS method for Large Scale Optimization, Mathematical Programming. 45. pp. 503-528.

[45] Galen Andrew, JianfengGao. 2007 Scalable Training of L_1-Regularized Log-Linear models, Microsoft Research, Redmond.

[46] JishaPJ. 2011. Parts-of-Speech Tagger and Chunker for Malayalam-Statistical approach, Computer Engineering and Intelligent Systems.