Morphology Based POS Tagging on Telugu

Srinivasu Badugu[1]

[1] **Department of Computer and Engineering, Sridevi Womens Engineering College
Hyderabad, Andhra Pradesh 500072, India**

Abstract

In this paper, we present a morphological based automatic tagging for Telugu without requiring any machine learning algorithm or training data. We believe that inflectional and agglutinating languages, the critical information required for tagging comes more from word internal structure than from the context and we show how a well designed morphological analyzer can assign correct tags and disambiguate many cases of tag ambiguities too. We have used fine grained, hierarchical tag set, carrying not only morph-syntactic information but also some aspects of lexical and semantic information that is necessary or useful for syntactic parsing. We give details of our experiments and results obtained. We believe our approach can also be applied to other Dravidian languages.

Keywords: **Morphology, POS tagging, Tagging, Telugu, Lexicon.**

1. Introduction

The ultimate goal of research on Natural language processing (NLP) is to understand human or natural languages and to facilitate human-machine interaction through human language or natural language. To achieve such research goal, NLP people has focused on different sub tasks. Part of speech tagging is one of such sub-task.

In computational linguistics part of speech tagging also called grammatical tagging is a classification system. Tagging is the process of assigning short labels to words in a text for the purpose of indicating lexical, morphological, syntactic, semantic or other such information associated with these words. When the focus is mainly on syntactic categories and/or sub-categories, this is also known as part-of-speech or POS tagging. It may be noted that the term tagging is broader than the term POS tagging. One of the main reasons for incorporating a tagging level between lexical and morphological levels on the one side and syntactic parsing on the other side is to reduce ambiguities. Tag ambiguities multiply at an exponential rate making syntactic parsing so much more difficult.

Word categories or classes are crucial to the study of sentence structure. In fact, they are more important than words. Each sentence has different words and different order. For example The Ram saw the running dear, and The running dear saw the Ram, and The Ram saw the dear running. Sentences having the same set of words can vary in meaning and the difference can only be accounted by the sentence structure.

Word classes such as noun, verb and adjective are also called 'Parts of Speech' (POS) by tradition. For the sake of convenience, we may use short labels, called tags, for these. For example, nouns may be indicated by N and verbs by V. POS Tagging are the process of attaching such short labels to indicate the Parts of Speech for words.

Tagging is only for convenience. However, tagging is usually intended to reduce, if not eliminate, ambiguities at word level. It is well known that syntactic parsing is at least cubic in computational complexity [19] and having to consider several alternative interpretations for each word can exponentially increase parsing complexity. Tagging has been invented in NLP as an independent layer of analysis, sitting between morphology and syntax, mainly to help the syntactic parser to do better in terms of speed. Therefore, syntactic parsing is actually orders of magnitude simpler than what we usually think it is. To this extent, the importance of tagging is reduced. It is worth noting that linguistic theories never posited tagging or chunking as separate layers of analysis sitting between morphology and syntax.

1.1 About Telugu

Telugu is a morphologically rich language resulting in its relatively free-word order characteristics. Normally most of Telugu words take more than one morphological suffix. Telugu nouns are inflected for number (singular, plural), gender (masculine, feminine, and neuter) and case (nominative, accusative, genitive, dative, vocative, instrumental, and locative). The principal parts of the verb morphology are the root, the infinitive, and the participles. There are three conjugations of Telugu verbs, each containing several classes of verbs. The five different verb forms (Present, Past, Future, and the Imperative, durative) are formed with the addition of personal affixes with some particles.

The relatively free word order of Telugu normally has the main verb in a terminating position and all other categories of words can occur in any position in the sentence. Pure word based approaches for POS tagging are not effective. Morphological analyzer can help for assigning POS categories of particular word.

Before we get into designing a tagger we must ask what is the purpose of tagging and where we get the information required to selecting a particular tag out of all the possible tags for a given word in a given sentence. The general assumption in most tagging work has been that this information comes from other words in the sentence. This is not always true. In the case of Dravidian, for example, we find that in most cases, information required for disambiguating tags comes from word internal structure, not from the other words in the sentential context. Morphology therefore does the major part of tagging and there is no need for Markov models and things of that kind. In fact we believe that the same approach can be effectively applied to all languages provided we change our view of what constitutes a word.

2. Approaches to Tagging

There are mainly two approaches for POS tagging 1) Linguistic or Rule-based approach 2) Machine learning or stochastic approach.

Rule-based tagging [6] is the oldest approach, in which there are two stages. In first stage, use a dictionary to assign all possible grammatical categories to each word. In second stage, use a large list of hand crafted rules to identify correct single tag for each ambiguous word. Disambiguation is done by analyzing the linguistic features of the word, its previous word, its following word and other aspects. For example, if the previous word is article then the next word must be noun. This information is coded in the form of rules.

The rule-based taggers are developed for European language. Rule based approaches for English achieved the best accuracy are 97.5 percent [6]. For Indian languages a rule based POS tagger for Tamil was developed and tested. It consists of 90 lexical rules and 7 context sensitive rules. It has given a precision of 92 percent [17].

Stochastic tagging techniques make use of corpus. The most common stochastic tagging uses a HMM (Hidden Markov Model). Stochastic tagging techniques can be either supervised/unsupervised/hybrid. In HMM the states usually denote the POS tags. The probabilities are estimated from a tagged training corpus or untagged training corpus in order to compute the most likely POS tags for the word of an input sentence. Stochastic tagging techniques can be of two types depending upon the training data.

Stochastic Supervised POS Tagging requires pre-tagged training corpus and uses HMM model whose parameters are calculated from tagged training corpus. In POS tagging task the input sentence is observed part, which is sequence of words. POS tags are represented as states of model. The states are hidden and we need to estimate state sequence for a given word sequence. We use viterbi algorithm for computing state sequence, which is most likely to generate the observed word sequence. POS tagging using HMM was developed for Bengali [25, 1].

Stochastic Unsupervised POS Tagging requires no tagged training corpus but instead use sophisticated computational methods to automatically induce tag sets, and based on these they calculate the probabilistic values needed by stochastic tagger.

Hybrid Model of POS Tagging, One way to achieve this is to have combination of both supervised and unsupervised methods. Other way of hybrid model is to have combination of supervised and rule based method. A hybrid POS tagger for Hindi, Bengali using supervised HMM technique and a rule based system was developed [23]. Other methods that are used for POS tagging for English are condition random fields, decision trees [27]. For Indian languages HMM based POS tagger were developed and tested by various people, but little amount of work is done in Telugu.

Machine learning approaches require training data. Generating training data is not an easy task and the quality and quantity may both be important considerations. Training data needs to be large and representative. Labeled training data can be either generated completely manually or tagged data generated by an existing tagger can be manually checked and refined to create high quality training data and both of these methods have their obvious limitations. In practice, we will have to live with sparse data and smoothing techniques used may introduce their own artifacts.

Given the limited amount of training data that is practically possible to develop, a large and detailed tag set will lead to sparsity of training data and machine learning algorithms will fail to learn effectively [7]. Manual tagging and checking also become difficult and error prone as the tag set becomes large and fine-grained and so there is a strong tendency to go for small, flat tag sets in machine learning approaches [12, 2, 3, 22 and 5]. Such small tag sets may not capture all the required and/or useful bits of information for carrying out syntactic parsing and other relevant tasks in NLP. Morphological features are essential for syntactic analysis in many cases. These have also been the conclusions of a practical experiment of using fine grained morphological tag set reported by Schmid and Laws [9]. Their experiments were carried out using German and Czech as examples of highly inflectional languages. Fine-grained distinctions may actually help to disambiguate other words in the local context. Flat tag sets are also rigid and resist changes. Hierarchical tag sets are more flexible. Thus the design of the tag-set is strongly influenced by the approach taken for tagging. Further, it is also influenced by the particular purpose for which tagging is taken up. A dependency parser of a particular kind may need a somewhat different sort of sub-categorization compared to, say, parsing using LFG or HPSG. Re-usability of tagged data across applications is an issue.

Although rule based approaches may appear to be formidable to start with, once the proper set of rules has been identified through a thorough linguistic study, there are many things to gain. Linguistic approaches can give us deeper and far-reaching insights into our languages and our mind. Knowledge based approaches generalize well, avoiding over-fitting, errors can be detected and corrected easily, improvements and refinements are easier too. In a pure machine learning approach, we can only hope to improve the performance of the system by generating larger and better training data and re-training the system, whereas in linguistic approaches, we can make corrections to the rules and guarantee the accuracy of tagging. Rule based approaches are also better at guessing and handling unknown words [28].

In this paper, we present an approach that does not depend upon statistical or machine learning techniques and there no need for any training data either. No manual tagging work is involved. We can afford to use a large, fine-grained, hierarchical tag set and still achieve high quality tagging automatically. We get both speed and accuracy. In this paper, we have chosen to render all Telugu words in Roman [16].

3. Morphology Based Tagging

3.1 Architecture

There is only one critical question that we need to ask when it comes to tagging – where can we find the crucial bits of information required to assign the correct tag to a given word in a given sentence? Statistical approaches assume that the necessary information comes from the other words in the sentence. In many cases, only the words that come before the current word are taken into direct consideration. We believe, in sharp contrast, that the crucial information required for assigning the correct tag comes from within the word. It is the internal structure of a word that determines its grammatical category as also sub-categorization and other features. True, there will be instances where the internal structure alone is not sufficient. Firstly we find that such cases are not as frequent as you may be thinking. A vast majority of the words can be tagged correctly by looking at the internal structure of the word. The crux of tagging lies in morphology. This is clearly true in the case of so called morphologically rich languages. Secondly, in those cases where morphology assigns more than one possible tag, information required for disambiguation comes mainly from syntax. Syntax implies complex inter-relationships between words and this cannot be reduced to a mere sequence of entities.

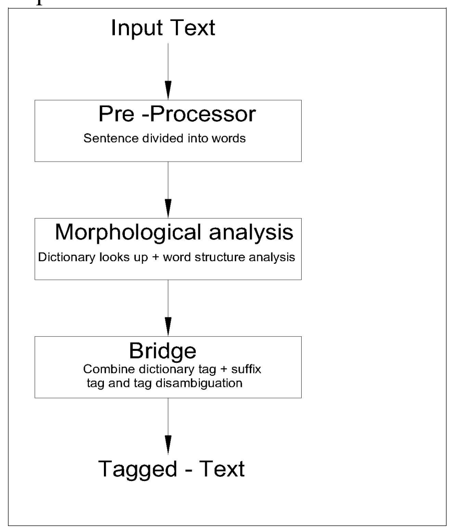


Fig. 1 The Tagging architecture

Statistical techniques are perhaps not the best means to capture and utilize such complex functional dependencies. Instead, chunking and parsing will automatically remove most of the tag ambiguities. Given this observation, we use simple pipe-line architecture as depicted in the figure below. We keep going forward and we do not need to come back again and again to preceding modules. We carry with us all the necessary/useful information in the form of tags, each module adding or refining the information as we move on. The pre-processing module performs several useful tasks but the most important task is to identify words correctly. We find that this is doable to a large extent in the case of Telugu at this stage itself. Where it is not feasible to divide given sentences into proper words, we proceed with morphology and after that, we will have sufficient and clear information required for obtaining proper words. This is one of the main tasks for the bridge module. Once we cross this morph-syntactic bridge, we can be sure that we are working only with proper words and their tags. The lexicon assigns tags to words that appear without any overt morphological inflection. Morphology handles all the derived and inflected words, including many forms of sandhi. The bridge module combines the tags given by the dictionary and the additional information given by the morph, making suitable changes to reflect the correct structure and meaning where required. The overall tag structure remains the same throughout, making it so much simpler and easier to build, test and use.

3.2 Morphology

English morphology is very simple and direct to implement. Morphological features also very few. The numbers of tags used for English POS tagging system are not that large: it ranges from 45 to 203 (in the case of CLAWS C8 tag-set) [21]. Also, average number of tags per token is low (2.32 tags per token on the manually tagged part of the Wall Street Journal corpus in the Penn Tree-bank) [13].The numbers of potential morphological tags in inflectional rich languages are theoretically unlimited [13]. In English many of the unknown words will be proper nouns but in inflectional and/or agglutinate languages such as Indian languages, many common nouns and verbs may be absent in the training corpus. Therefore, a good morphological analyzer helps [11, 18 and 7].
POS tagging for English seems to have reached the top level, but full morphological tagging for inflectionally rich languages such as Romanian, Hungarian, is still an open problem [13]. Indian Languages are highly inflectional and agglutinating too.

Statistical approaches assume that the information necessary for tag assignment comes from the other tokens in the sentence. In many cases, only the tokens that come before the current word are taken into direct consideration. We believe, in sharp contrast, that the crucial information required for assigning the correct tag comes from within the word, in Dravidian languages. The crux of tagging lies in morphology.

In Telugu, as in many other languages, morphological inflections apply mainly to nouns and verbs. Adjectives and adverbs show little or no inflection. Ambiguity between noun and verb is the most problematic issue in parsing. Nouns and verbs occur mostly in inflected form in Telugu. Further, noun and verb morphology are generally mostly or completely disjoint. Therefore, morphology can resolve most of the critical tag ambiguities. The remaining ambiguities can be resolved using well defined local syntactic rules. Even after all this, if a very small amount of ambiguity remains let it remain, there is no harm. Syntactic parsers must anyway have the capacity to deal with ambiguities. Thus, in our approach, morphology plays a major role in tag assignment as also in tag disambiguation.

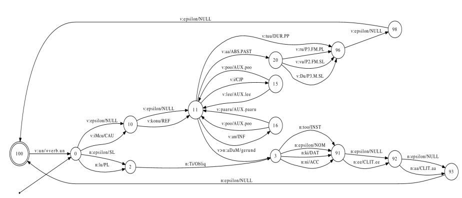
The dictionary lists the root forms or the base forms of words in a given language and defines the meaning of each. When a listener hears an inflected or derived form, how exactly does he or she understand the meaning? Morphology is that part of grammar that systematically relates the internal structure of words to the meaning of the whole word form. While lexical meaning comes from the dictionary, the meaning changes brought about by the affixes is to be determined by morphology. For example, if a Telugu person hears the word 'cestaanu' ((he) did (something)), the listener knows that the root word 'ceeyu' means 'do'. How exactly does he come to know that it is one single masculine person, other than the speaker and listener, who did something, and how exactly does he come to know that the action refers to a past event? This explanatory capability is the crux of morphology. Morphology should be useful for teaching and learning the language, for gaining deeper insights into how the language works.

The morph system is implemented as an extended Finite State Transducer. The FST has 398 transitions or arcs. The figure below shows a small part of the FST. A category field has been incorporated so that only relevant transitions are allowed. Derivation is handled by allowing category changes. Transitions are on morphemes, not on individual characters or letters. Dravidian morphology involves complex morpho-phonemic changes at the juncture of morphemes and linguistically motivated rules have been used to handle these [4].

We find that in any running text approximately 40% of the words are found directly in the dictionary. Less than 2% of the words in the dictionary are ambiguous. About one third of these are ambiguous between noun and verb. Since nominal and verbal morphology are more or less completely disjoint in Telugu, and since these words occur mostly in inflected forms (more than 92% of times), morphology can resolve most of these cases of ambiguity. Morphology can also resolve ambiguity between nouns / verbs other categories such as adjectives and adverbs.

Thus, morphology has a very important role in tagging. If we work with proper words instead of tokens, we believe we will get a similar picture in other inflectional languages. Certain kinds of systematic structural ambiguities in a language can lead to multiple tag assignments, calling for further disambiguation.

Fig. 2 Sample FSM for Morphological analysis



3.3 Designing a Tag-Set

The design of a tag-set is critically dependent on the purpose and the approach taken for tagging. The beaten path is to develop a manually tagged database of sentences and then use this for training a machine

IJCSI International Journal of Computer Science Issues, Vol. 11, Issue 1, No 1, January 2014
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org

learning algorithm [26, 23, 1, 10, 20 and 24]. The machine learning algorithm is expected to generalize from these training examples so that it can then tag any new sentence. Manual tagging is difficult, time consuming and prone to human errors. Consistency is difficult to achieve especially if the tag set is fine grained and elaborate. Also, given the limited amount of training data that is practically possible to develop, a large and detailed tag set will lead to sparsity of training data and machine learning algorithms will fail to learn effectively[7, 9]. From these considerations, researchers tend to restrict themselves to small, shallow or flat tag sets which are also least confusing to human annotators [23]. When this idea is taken to the extreme, useful information may be lost. Flat tag sets are also rigid and resist changes. Hierarchical tag sets are more flexible. In this paper we propose an alternative view and a novel approach to tagging. We do not depend upon statistical or machine learning techniques and we do not need any training data. No manual tagging work is involved and so we can afford to use a large, fine grained, hierarchical tag set that carries a lot of lexical, morphological, syntactic and semantic information.

One of the biggest difficulties that researchers face while designing tag sets, while performing manual tagging and while building and evaluating tagging systems is the very definition of tags. Tags involve lexical, morphological, syntactic and semantic considerations and often there are conflicts. One cannot go purely by intuitive definitions such as 'nouns are things, pronouns stand in place of nouns and adjectives modify nouns'. We will need to give precise definitions and criteria to decide which tag label should be given to which word. While meaning is supreme in language, grammatical considerations dictate the fine details. We will need to sub-categorize the major categories to reflect the significant differences these sub-categories exhibit in syntax and these sub-categories also need to be defined very precisely. We have developed a detailed, hierarchical tag set keeping these issues in mind. See [15] for full details.

We developed Telugu lexicon using hierarchical tag-set, currently there are 52,351 entries in our Telugu Lexicon system. Of these, only 649 are ambiguous. N-COM-COU-N.SL-NOM, PRO-PER-P3.N.SL-PROX-NOM, N-CARD-NHU-N.SL-DAT, V-TR1 are examples of tags we use. We call the complete labels such as N-COM-COU-N.SL-NOM as single tags. Tags are made up of tag elements such as NOM and N.SL. Tag elements may in turn be made up of tag atoms. N.SL is one tag element with two tag atoms. The first field is always the main category and the subsequent one or two fields may indicate sub-categories. Rest of the fields indicates grammatical features. There are 270 unique tags in the dictionary, there are 138 unique tag elements and 121 unique tag atoms.

Table 1: Dictionary Tag-set

N (NOUN) COM(Common) CARD(Cardinal) LOC (locative) PRP(Proper) - PER(Personal) -LOC(Location) -ORG(Organ.) -OTH(Others)	ADV (Adverbs) MAN(Manner) CONJ (conjunction) PLA(Place) TIM(Time) QW(Question word) INTF(Intensifier) POSN(Post - Nominal Modifier) ABS (Absolute)
PRO (Pronoun) PER(Personal) INTG(Interrogative) REF (reflexive) INDF (indefinite)	CONJ (Conjunction) COOR(Coordinating) SUB(Subordinating)
ADJ (Adjective) DEM(Demonstrative) QNTF(Quantifying) ORD(Ordinal) ABS(Absolute) QW(Question word)	V (Verb) IN(Intransitive) TR(Transitive) BI(Bi-transitive) DEFE(defective)
SYMB(Symbol)&	INTJ (Interjection)

Here are some examples of tags in the dictionary.

peddagaa||ADV-MAN muduru||ADJ-ABS||V-IN

aMdamaina||ADJ-ABS telusu||V-DEFE
tinu||V-TR paatika||N-CARD-NHU-NOM

baDi||N-COM-COU-N.SL-NOM

adhikaari||N-COM-COU-FM.SL-NOM

ataDu||PRO-PER-P3.M.SL-DIST-NOM

3.4 Bridge

The lexicon assigns tags to words that appear without any overt morphological inflection. Morphology handles all the derived and inflected words, including many forms of sandhi. The bridge module combines the tags given by the dictionary and the additional information given by the morph, ensuring that the correct structure (and hence meaning) are depicted by the tags. The overall tag structure remains the same throughout, making it so much simpler and easier to build, test and use.

Let us now look at some examples of the output of morphological analyzer.

ceppu have to meanings 1) to say or tell 2) a shoe or slipper

ceppaaDu (he told) :

ceppaaDu<ceppu:N-COM-COU-N.SL-NOM||V-TR12:
%v-ABS.PAST-P3.M.SL-%-->

ceppuku (to the shoe) :
ceppuku<ceppu:N-COM-COU-N.SL-NOM||V-TR12: %n-SL-obliq-DAT>

ceppinavaaDu (the one who said):
ceppinavaaDu <ceppu:N-COM-COU-N.SL-NOM||V-TR12:%v-POST.RP-%adj-PRON.vaaDu.P3.M.SL-%n-NOM-%-->

cepputoo (with the shoe):
cepputoo<ceppu:N-COM-COU-N.SL-NOM||V-TR12:%n-SL-obliq-INST-%-->

ceppulanu (to the shoes):
ceppulanu<ceppu:N-COM-COU-N.SL-NOM||V-TR12:%n-PL-obliq-ACC-%-->

Here the bits of information obtained from the dictionary and morphology are combined to generate final tags. For the examples shown above, the tagger will produce the following tags

ceppaaDu (he told):
ceppaaDu||ceppu||V-TR12-ABS.PAST-P3.M.SL

ceppuku(to the shoe):
ceppuku||ceppu||N-COM-COU-N. SL-DAT

ceppinavaDu (the one who told):
ceppinavaaDu||ceppu||V-TR12.v-POST.RP-.adj-PRON.
aaDu.P3.M.SL-.n-NOM

cepputoo(with the shoe):
cepputoo||ceppu||N-COM-COU-N.SL-INST

ceppulanu(to the shoes):
ceppulanu||ceppu||N-COM-COU-N.PL-ACC

4. Experiments and Results

There are no publicly available standard data sets available for Telugu. We have developed our own Telugu text corpus of about 50 Million words [8]. We have tested our system on a corpus of 15 Million words. Performance of the morph analyzer on randomly selected sentences from this corpus is shown below:

Table 2: Performance of Morphological analyzer

File name	# words	#tagged	#fully correct	P (precision)	R (Recall)	F (F-measure)
F1	861	838	831	99.16	96.52	97.82
F2	4910	4575	4543	99.30	92.53	95.79
F3	9269	8560	8502	99.32	91.72	95.37
F4	14092	12965	12850	99.11	91.19	94.99
Total	29010	26691	26471	99.18	91.25	95.05

It is generally found that in any running text about 40% of the word forms are directly found in the dictionary and the rest are analyzed by morph [8]. Only some 5 to 10% of the words are ambiguous. Of these ambiguous words, 50 to 60% are uninflected and these ambiguities cannot be resolved by morph. A vast majority of structurally ambiguous words are resolved by morph. Most of the remaining ambiguities can be resolved using local context within the Bridge module. Here local syntactic constraints or chunking rules are used to resolve the ambiguities. Only a very small percentage of words will remain ambiguous after tagging. In most cases, ambiguity is restricted to two tags. Most of these remaining cases of ambiguity are cases of true inherent ambiguity, that is, where the words have more than one meaning and sentences containing them may also be ambiguous in meaning. Some 5 to 10% of words will remain untagged. It is found that most of these words are loan words, named entities and compounds/words involving external sandhi. Efforts are on to improve the system along these dimensions. Since the whole system is rule governed, the results can be guaranteed to be correct. Manual verifications have validated this claim. In summary, we can say our system is capable of fairly high performance tagging.

5. Conclusions

In this paper we have presented a new approach to tagging using a morphological analyzer and a fine-grained hierarchical tag-set. We have shown that it is possible to develop high performance tagging system without need for any training data or machine learning or statistical inference for any training data or machine learning or statistical inference. Since the whole system is rule governed, the results can be guaranteed to be correct. Manual verification has validated this claim.

References

[1] Asif Ekbal and Samiran Mandal, "POS Tagging using HMM and Rule based Chunking", In Proceedings of International Joint Conference on Artificial Intelligence Workshop on Shallow Parsing for South Asian Languages, IIIT-Hyderabad, Hyderabad, India, 2007.
[2] "AU-KBC POS Tagset for Tamil", AU-KBC, Anna University,http://nrcfosshelpline.in/smedia/images/downloads/Tamil_Tagset-opensource.odt
[3] Baskaran Sankaran, Kalika Bali, Monojit Choudhury, Tanmoy Bhattacharya, Pushpak Bhattacharyya, Girish Nath Jha, S. Rajendran, K. Saravanan, L. Sobha, and K.V.

Subbarao, "A Common Parts-of-Speech Tagset Framework for Indian Languages", In Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco, 2008, pp 1331–1337.

[4] BH. Krishnamurthi and J.P.L Gwynn, "A Grammar of Modern Telugu", Oxford University Press, New Delhi, 1985.

[5] David Elworthy, "Tagset Design and Inflected Languages", In In EACL SIGDAT workshop From Texts to Tags: Issues in Multilingual Language Analysis, 1995, pp 1–10.

[6] Eric Brill, "A Simple Rule based PoS Tagger", In Proceedings of Third Annual Conference on Applied Natural Language Processing, Trento, Italy, 1992.

[7] ES Atwell, "Development of Tag Sets for Part-of-Speech Tagging", In Ludeling A and Kyto M,editors, Corpus Linguistics: An International Handbook, Mouton de Gruyter, 2008, pp 501–526.

[8] G Bharadwaja Kumar, Kavi Narayana Murthy, and B B Chaudhuri, "Statistical Analysis of Telugu Text Corpora", In International Journal of Dravidian Languages, Vol. 36, No. 2, June 2007, pp. 71–99.

[9] Helmut Schmid and Florian "Laws Estimation of Conditional Probabilities With Decision Trees and an Application to Fine-Grained POS Tagging", In COLING, 2008, pp 777–784.

[10] Himanshu Agarwal and Anirudh Mani, "Part of Speech Tagging and Chunking with Conditional Random Fields", In Proceedings of NLPAI Ma-chine Learning Workshop on Part of Speech Tagging and Chunking for Indian languages, IIIT-Hyderabad, Hyderabad, India, 2006.

[11] Huihsin Tseng, Daniel Jurafsky, and Christopher Manning, "Morphological Features help POS Tagging of unknown Words across Language Varieties", In Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, Jeju Island, Korea, October 2005, pp 32–39.

[12] IIIT-Hyderabad, "A Part-of-Speech Tagset for Indian Languages"IIIT-Hyderabad, http://shiva.iiit.ac.in /SPSAL 20 07 iiit_tagset_ guidelines.pdf".

[13] Jan Haji, "Morphological Tagging: Data vs. Dictionaries", In Proceedings of the 6th Applied Natural Language Processing and the 1st NAACL Conference, Seattle, Washington, 2000, pp 94–101.

[14] J Steven and DeRose, "Grammatical Category Disambigua-tion by Statistical Optimization", Computational Linguistics, Vol. 14, No. 1, 1988, pp 31–39.

[15] Kavi Narayana Murthy and Badugu Srinivasu, "On the Design of a TagSet for Dravidian Languages", In 40th All India Conference of Dravidian Linguists, University of Hyderabad, Hyderabad, India, 18-20 JUNE-2012.

[16] Kavi Narayana Murthy and Badugu Srinivasu,"Roman Transliteration of Indic Scripts", In 10th International Conference on Computer Applications, University of Computer Studies, Yangon, Myanmar, February 2012, pp. 28-29.

[17] L Sobha P Arulmozhi, "Tagger for a Relatively Free Word Order Language", journal of Research in Computing Science, Vol. 8, February 2006, pp. 75–88.

[18] Majdi Sawalha and Eric Atwell, "Fine-grain Morphological Analyzer and Part-of-Speech Tagger for Arabic Text", In Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta, 2010, pp 1258–1265.

[19] Mary Dalrymple, "How much can Part-of-Speech Tagging help Parsing?", Natural Language Engineering, 2006, Vol. 12, No.4, pp. 373–389.

[20] Pranjal Awasthi, Delip Rao and Balaraman Ravindran, "Part of Speech Tagging and Chunking with HMM and CRF", In Proceedings of NLPAI Machine Learning Workshop on Part of Speech Tagging and Chunking for Indian languages, IIIT-Hyderabad, Hyderabad, India, 2006.

[21] R Garside, "The CLAWS Word-Tagging System", The Computational Analysis of English, Longman, 1987, pp. 30–41.

[22] R. J. Rama Sree, G Umamaheshwar Rao, and K. V Madhu Murthy, "Assessment and Development of POS Tagset for Telugu", In Proceedings of the Sixth Workshop on Asian Language Resources, 3rd International Joint Conference on Natural Language Processing (IJCNLP-08), IIIT Hyderabad, Hyderabad, India, 2008, pp 85–88.

[23] R K Rao Pattabhi, R Vijay Sundar Ram, R Vijay Krishna, and L Sobha, "A Text Chunker and Hybrid POS Tagger for Indian Languages", In Proceedings of International Joint Conference on Artificial Intelligence Workshop on Shallow Parsing for South Asian Languages, IIIT-Hyderabad, Hyderabad, India, 2007.

[24] Sandipan Dandapat and Sudeshna Sarkar, "Part of Speech Tagging for Bengali with Hidden Markov Model", In Proceedings of NLPAI Machine Learning Workshop on Part of Speech Tagging and Chunking for Indian languages, IIIT-Hyderabad, Hyderabad, India, 2006.

[25] Sandipan Dandapat, Sudeshna Sarkar, and Anupam Basu, "A Hybrid Model for Part of Speech Tagging and its Application to Bengali" In Proceedings of International Conference on Computational Intelligence, Instanbul, Turkey, 2004, pp 169–172.

[26] Sankaran Baskaran, "Hindi Part of Speech Tagging and Chunking", In Proceedings of NLPAI Machine Learning Workshop on Part of Speech Tagging and Chunking for Indian languages, IIIT Hyderabad, Hyderabad, India, 2006.

[27] Sathish Chandra Pammi and Kishore Prahlad, "PoS Tagging and Chunking using Decision Forests", In Proceedings of International Joint Conference on Artificial Intelligence workshop on Shallow Parsing for South Asian Languages, Language Technologies Research Centre, IIIT-Hyderabad, 2007.

[28] Steven Abney, "Part-of-Speech Tagging and Partial Parsing", In Corpus Based Methods in Language and Speech, Kluwer Academic Publishers, 1996, pp 118–136.

.