Morphological Richness Offsets Resource Demand- Experiences in Constructing a POS Tagger for Hindi

Smriti Singh **Kuhoo Gupta** **Manish Shrivastava** **Pushpak Bhattacharyya**
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Powai, Mumbai
400076 Maharashtra, India
{smriti,kuhoo,manshri,pb}@cse.iitb.ac.in

Abstract

In this paper we report our work on building a POS tagger for a morphologically rich language- Hindi. The theme of the research is to vindicate the stand that- if morphology is strong and harnessable, then lack of training corpora is not debilitating. We establish a methodology of POS tagging which the resource disadvantaged (lacking annotated corpora) languages can make use of. The methodology makes use of locally annotated modestly-sized corpora (15,562 words), exhaustive morpohological analysis backed by high-coverage lexicon and a decision tree based learning algorithm (CN2). The evaluation of the system was done with 4-fold cross validation of the corpora in the news domain (www.bbc.co.uk/hindi). The current accuracy of POS tagging is 93.45% and can be further improved.

1 Motivation and Problem Definition

Part-Of-Speech (POS) tagging is a complex task fraught with challenges like *ambiguity of parts of speech* and *handling of "lexical absence" (proper nouns, foreign words, derivationally morphed words, spelling variations* and *other unknown words)* (Manning and Schutze, 2002). For English there are many POS taggers, employing machine learning techniques like transformation-based error-driven learning (Brill, 1995), decision trees (Black et al., 1992), markov model (Cutting et al. 1992), maximum entropy methods (Ratnaparkhi, 1996) *etc*. There are also taggers which are hybrid using both stochastic and rule-based approaches, such as CLAWS (Garside and Smith, 1997). The accuracy of these taggers ranges from 93-98% approximately. English has annotated corpora in abundance, enabling usage of powerful data driven machine learning methods. But, very few languages in the world have the resource advantage that English enjoys.

In this scenario, POS tagging of highly inflectional languages presents an interesting case study. Morphologically rich languages are characterized by a large number of morphemes in a single word, where morpheme boundaries are difficult to detect because they are fused together. They are typically free-word ordered, which causes fixed-context systems to be hardly adequate for statistical approaches (Samuelsson and Voutilainen, 1997). Morphology-based POS tagging of some languages like Turkish (Oflazer and Kuruoz, 1994), Arabic (Guiassa, 2006), Czech (Hajic et al., 2001), Modern Greek (Orphanos et al., 1999) and Hungarian (Megyesi, 1999) has been tried out using a combination of hand-crafted rules and statistical learning. These systems use large amount of corpora along with morphological analysis to POS tag the texts. It may be noted that a purely rule-based or a purely stochastic approach will not be effective for such

languages, since the former demands subtle linguistic expertise and the latter variously permuted corpora.

1.1 Previous Work on Hindi POS Tagging

There is some amount of work done on morphology-based disambiguation in Hindi POS tagging. Bharati *et al.* (1995) in their work on computational Paninian parser, describe a technique where POS tagging is implicit and is merged with the parsing phase. Ray *et al.* (2003) proposed an algorithm that identifies Hindi word groups on the basis of the lexical tags of the individual words. Their partial POS tagger (as they call it) reduces the number of possible tags for a given sentence by imposing some constraints on the sequence of lexical categories that are possible in a Hindi sentence. UPENN also has an online Hindi morphological tagger[1] but there exists no literature discussing the performance of the tagger.

1.2 Our Approach

We present in this paper a POS tagger for Hindi- the national language of India, spoken by 500 million people and ranking 4th in the world. We establish a methodology of POS tagging which **the resource disadvantaged (lacking annotated corpora) languages can make use of**. This methodology uses locally annotated modestly sized corpora (15,562 words), exhaustive morphological analysis backed by high-coverage lexicon and a decision tree based learning algorithm- CN2 (Clark and Niblett, 1989). To the best of our knowledge, such an approach has never been tried out for Hindi. The heart of the system is the detailed linguistic analysis of morphosyntactic phenomena, adroit handling of suffixes, accurate verb group identification and learning of disambiguation rules.

The approach can be used for other inflectional languages by providing the language specific resources in the form of suffix replacement rules (SRRs), lexicon, group identification and morpheme analysis rules *etc* and keeping the processes the same as shown in Figure 1. The similar kind of work exploiting morphological information to assign POS tags is under progress for Marathi which is also an Indian language.

In what follows, we discuss in section 2 the challenges in Hindi POS tagging followed by a section on morphological structure of Hindi. Section 4 presents the design of Hindi POS tagger. The experimental setup and results are given in sections 5 and 6. Section 7 concludes the paper.

2 Challenges of POS Tagging in Hindi

The inter-POS ambiguity surfaces when a word or a morpheme displays an ambiguity across POS categories. Such a word has multiple entries in the lexicon (one for each category). After stemming, the word would be assigned all possible POS tags based on the number of entries it has in the lexicon. The complexity of the task can be understood looking at the following English sentence where the word 'back' falls into three different POS categories-

"I get back to the back seat to give rest to my back."

The complexity further increases when it comes to tagging a free-word order language like Hindi where almost all the permutations of words in a clause are possible (Shrivastava et al., 2005). This phenomenon in the language, makes the task of a stochastic tagger difficult.

Intra-POS ambiguity arises when a word has one POS with different feature values, *e.g.*, the word 'लड़के' {laDke} *(boys/boy)* in Hindi is a noun but can be analyzed in two ways in terms of its feature values:

1. **POS**: *Noun*, **Number**: *Sg*, **Case**: *Oblique*
   मैंने लड़के को एक आम दिया.
   maine laDke ko ek aam diyaa.
   I-erg boy to one mango gave.
   I gave a mango to the boy.

2. **POS**: *Noun*, **Number**: *Pl*, **Case**: *Direct*
   लड़के आम खाते हैं.
   laDke aam khaate hain.
   Boys mangoes eat.
   Boys eat mangoes.

[1] http://ccat.sas.upenn.edu/plc/tamilweb/hindi.html

One of the difficult tasks here is to choose the appropriate tag based on the morphology of the word and the context used. Also, new words appear all the time in the texts. Thus, a method for determining the tag of a new word is needed when it is not present in the lexicon. This is done using context information and the information coded in the affixes, as affixes in Hindi (especially in nouns and verbs) are strong indicators of a word's POS category. For example, it is possible to determine that the word 'जाएगा' {jaaegaa} *(will go)* is a verb, based on the environment in which it appears and the knowledge that it carries the inflectional suffix -एगा {egaa} that attaches to the base verb 'जा' {jaa}.

2.1 Ambiguity Schemes

The criterion to decide whether the tag of a word is a *Noun* or a *Verb* is entirely different from that of whether a word is an *Adjective* or an *Adverb*. For example, the word 'पर' can occur as *conjunction*, *post-position* or a *noun* (as shown previously), hence it falls in an *Ambiguity Scheme* '*Conjunction-Noun-Postposition*'. We grouped all the ambiguous words into sets according to the *Ambiguity Schemes* that are possible in Hindi, e.g., *Adjective-Noun*, *Adjective-Adverb*, *Noun-Verb*, *etc*. This idea was first proposed by Orphanos *et al.* (1999) for Modern Greek POS tagging.

3 Morphological Structure Of Hindi

In Hindi, *Nouns* inflect for number and case. To capture their morphological variations, they can be categorized into various *paradigms*[2] (Narayana, 1994) based on their *vowel ending, gender, number* and *case information*. We have a list of around 29,000 Hindi nouns that are categorized into such paradigms[3]. Looking at the morphological patterns of the words in a paradigm, suffix-replacement rules have been developed. These rules help in separating out a valid suffix from an inflected word to output the correct stem and consequently, get the correct root.

Hindi *Adjectives* may be inflected or uninflected, *e.g.*, 'चमकीला' {chamkiilaa} *(shiny)*, 'अच्छा' {acchaa} *(nice)*, 'लंबा' {lambaa} *(long)* inflect based on the number and case values of their head nouns while 'सुंदर' {sundar} *(beautiful)*, 'भारी' {bhaarii} *(heavy) etc*. do not inflect.

Hindi *Verbs* inflect for the following grammatical properties (GNPTAM):

1. Gender: Masculine, Feminine, Non-specific
2. Number: Singular, Plural, Non-specific
3. Person: 1st, 2nd and 3rd
4. Tense: Past, Present, Future
5. Aspect: Perfective, Completive, Frequentative, Habitual, Durative, Inceptive, Stative
6. Modality: Imperative, Probabilitive, Subjunctive, Conditional, Deontic, Abilitive, Permissive

The morphemes attached to a verb along with their corresponding analyses help identify values for GNPTAM features for a given verb form.

Division of Information Load in Hindi Verb Groups

A *Verb Group* (VG) primarily comprises main verb and auxiliaries. Constituents like particles, negation markers, conjunction, *etc*. can also occur within a VG. It is important to know how much of GNPTAM feature information is stored in VG constituents individually and what is the load division in the absence or presence of auxiliaries. In a Hindi VG, when there is no auxiliary present, the complete information load falls on the main verb which carries information for GNPTAM features. In presence of auxiliaries, the load gets shared between the main verb and auxiliaries, and is represented in the form of different morphemes (inflected or uninflected), *e.g.*, in the sentence -

[2] A paradigm systematically arranges and identifies the uninflected forms of the words that share similar inflectional patterns.

[3] Anusaaraka system developed at IIT Kanpur (INDIA) uses similar noun sets in the form of paradigms

मैं बोल पा रहा हूँ
main bol paa rahaa hoon
I am able to speak

1. Main verb 'बोल' {bol} is uninflected and does not carry any information for any of the GNPTAM features.

2. 'पा' {paa} is uninflected and gives modality information, *i.e.*, Abilitive.

3. 'रहा' {rahaa} gives Number (Sg), Gender (Masculine), Aspect (Durative)

4. 'हूँ' {hoon} gives Number (Sg), Person (1st), Tense (Present)

Gerund Identification
In Hindi, the attachment of verbal suffixes like 'ना' {naa} and 'ने' {ne} to a verb root results either in a gerund like 'तैरना' {tairnaa} *(swimming)* or in an infinitival verb form like 'तैरना' {tairnaa} *(to swim)*. We observed that it is easy to detect a gerund if it is followed by a case-marker or by any other infinitival verb form.

4 Design of Hindi POS Tagger

4.1 Morphology Driven Tagger

Morphology driven tagger makes use of the affix information stored in a word and assigns a POS tag using no contextual information. Though, it does take into account the previous and the next word in a VG to correctly identify the main verb and the auxiliaries, other POS categories are identified through lexicon lookup of the root form. The current lexicon[4] has around 42,000 entries belonging to the major categories as mentioned in Figure 3. The format of each entry is ⟨word⟩,⟨paradigm⟩,⟨category⟩.

The process does not involve learning or disambiguation of any sort and is completely driven by hand-crafted morphology rules. The architecture of the tagger is shown in Figure 1. The work progresses at two levels:

[4]The lexicon was developed using the wordlist from Hindi Wordnet (http://www.cfilt.iitb.ac.in/wordnet/webhwn/) and partial noun list from Anusaraka. It is being enhanced by adding new words from the corpus and removing the inconsistencies.

1. **At Word Level**: A *stemmer* is used in conjunction with lexicon and Suffix Replacement Rules (SRRs) to output all possible root-suffix pairs along with POS category label for a word. There is a possibility that the input word is not found in the lexicon and does not carry any inflectional suffix. In such a case, *derivational morphology rules* are applied.

2. **At Group Level**: At this level a *Morphological Analyzer* (MA) uses the information encoded in the extracted suffix to add morphological information to the word. For nouns, the information provided by the suffixes is restricted only to '*Number*'. '*Case*' can be inferred later by looking at the neighbouring words.

   For verbs, GNP values are found at the word level, while TAM values are identified during the VG Identification phase, described later. The analysis of the suffix is done in a discrete manner, *i.e.*, each component of the suffix is analyzed separately. A morpheme analysis table comprising individual morphemes with their paradigm information and analyses is used for this purpose. MA's output for the word खाऊंगी {khaaoongii} *(will eat)* looks like -
   Stem: खा *(eat)*
   Suffix: ऊंगी Category: *Verb*
   Morpheme 1: ऊं Analysis: *1 Per, Sg*
   Morpheme 2: ग Analysis: *Future*
   Morpheme 3: ई Analysis: *Feminine*

4.1.1 Verb Group Identification

The structure of a Hindi VG is relatively rigid and can be captured well using simple syntactic rules. In Hindi, certain auxiliaries like 'रह' {rah}, 'पा' {paa}, 'सक', {sak} or 'पड़' {paD} can also occur as main verbs in some contexts. VG identification deals with identifying the main verb and the auxiliaries of a VG while discounting for particles, conjunctions and negation markers. The VG identification goes left to right by marking the first constituent as the main verb or copula verb and making every other verb con-

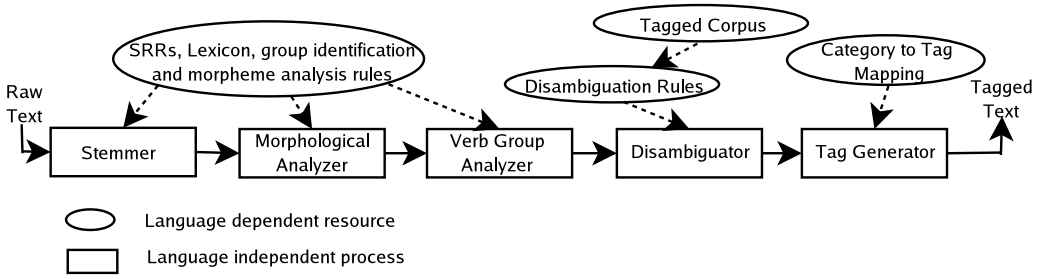

Figure 1: Overall Architecture of the Tagger

Table 1: Average Accuracy(%) Comparison of Various Approaches

LLB	LLBD	MD	BL	LB
61.19	86.77	73.62	82.63	93.45

struct an auxiliary till a non-VG constituent is encountered. Main verb and copula verb can take the head position of a VG and can occur with or without auxiliary verbs. Auxiliary verbs, on the other hand, always come along with a main verb or a copula verb. This results in a very high accuracy of 99.5% for verb auxiliaries. Ambiguity between a main verb and a copula verb remains unresolved at this level and asks for disambiguation rules.

4.2 Need for Disambiguation

The accuracy obtained by simple lexicon lookup based approach (LLB) comes out to be 61.19%. The morphology-driven tagger, on the other hand, performs better than just lexicon lookup but still results in considerable ambiguity. These results are significant as they present a strong case in favor of using detailed morphological analysis. Similar observation has been presented by Uchimoto *et al.* (2001) for Japanese language. According to the tagging performed by SRRs and the lexicon, a word receives *n* tags if it belongs to *n* POSs. If we consider multiple tags for a word as an error of the tagger (even when the options contain the correct tag for a word), then the accuracy of the tagger comes to be 73.62% (as shown in Table 1). The goal is to keep the contextually appropriate tag and eliminate others which can be achieved by devising a disambiguation technique. The disambiguation task can be naively addressed by choosing the most frequent tag for a word. This approach is also known as baseline (BL) tagging. The baseline accuracy turns out to be 82.63% which is still higher than that of the morphology-driven tagger[5]. The drawback with baseline tagging is that its accuracy cannot be further improved. On the other hand, there is enough room for improving upon the accuracy of morphology-driven (MD) tagger. It is quite evident that though the MD tagger works well for VG and many close categories, around 30% of the words are either ambiguous or unknown. Hence, a disambiguation stage is needed to shoot up the accuracy.

The common choice for disambiguation rule learning in POS tagging task is usually machine learning techniques mainly focussing on decision tree based algorithms (Orphanos and Christodoulalds, 1999), neural networks (Schmid, 1994), *etc*. Among the various decision tree based algorithms like ID3, AQR, ASSISTANT and CN2, CN2 is known to perform better than the rest (Clark and Niblett, 1989). Since no such machine learning technique has been used for Hindi language, we thought of choosing CN2 as it performs well on noisy data[6].

[5]These numbers may change if we experiment on a different dataset

[6]The training annotated corpora becomes noisy by virtue of intuitions of different annotators (trained native Hindi speakers)

4.2.1 Training Corpora

We set up a corpus, collecting sentences from BBC news site[7] and let the morphology-driven tagger assign morphosyntactic tags to all the words. For an ambiguous word, the contextually appropriate POS tag is manually chosen. Unknown words are assigned a correct tag based on their context and usage.

4.2.2 Learning

Out of the completely manually corrected corpora of 15,562 tokens, we created training instances for each *Ambiguity Scheme* and for *Unknown* words. These training instances take into account the POS categories of the neighbouring words and not the feature values[8]. The experiments were carried out for different context window sizes ranging from 2 to 20 to find the best configuration.

4.2.3 Rule Generation

The rules are generated from the training corpora by extracting the ambiguity scheme (AS) of each word. If the word is not present in the lexicon then its AS is set as 'unknown'. Once the AS is identified, a training instance is formed. This training instance contains the neighbouring correct POS categories as attributes. The number of neighbours included in the training instance is the window size for CN2. After all the ambiguous words are processed and training instances for all seen ASs are created, the CN2 algorithm is applied over the training instances to generate actual rule-sets for each AS. The CN2 algorithm gives one set of *If-Then* rules (either ordered or unordered) for each AS including 'unknown'[9]. The AS of every ambiguous word is formed while tagging. A corresponding rule-set for that AS is then identified and traversed to get the contextually appropriate rule. The resultant category outputted by this rule is then assigned to the ambiguous word. The traversal rule differs for ordered and unordered implementation. The POS of an unknown word is guessed by traversing the rule-set for unknown words[10] and assigning it the resultant tag.

[7] http://www.bbc.co.uk/hindi/

[8] Considering that a tag encodes 0 to 6 morphosyntactic features and each feature takes either one or a disjunction of 2 to 7 values, the total number of different tags can count up to several hundreds

[9] We used the CN2 algorithm implementation (1990) by Robin Boswell. The software is available at *ftp://ftp.cs.utexas.edu/pub/pclark/cn2.tar.Z*

5 Experimental Setup

The experimentation involved, first, identifying the best parameter values for the CN2 algorithm and second, evaluating the performance of the disambiguation rules generated by CN2 for the POS tagging task.

5.1 CN2 Parameters

The various parameters in CN2 algorithm are: rule type (ordered or unordered), star size, significance threshold and size of the training instances (window size). The best results are empirically achieved with ordered rules, star size as 1, significance threshold as 10 and window size 4, *i.e.*, two neighbours on either side are used to generate the training instances.

5.2 Evaluation

The tests are performed on contiguous partitions of the corpora (15,562 words) that are 75% training set and 25% testing set.

$$Accuracy = \frac{no.\ of\ single\ correct\ tags}{total\ no.\ of\ tokens}$$

The results are obtained by performing a 4-fold cross validation over the corpora. Figure 2 gives the learning curve of the disambiguation module for varying corpora sizes starting from 1000 to the complete training corpora size. The accuracy for known and unknown words is also measured separately.

6 Results and Discussion

The average accuracy of the learning based (LB) tagger after 4-fold cross validation is 93.45%. To

[10] Most of the unknown words are proper nouns, which cannot be stored in the lexicon extensively. So, it also helps in *named-entity detection*.

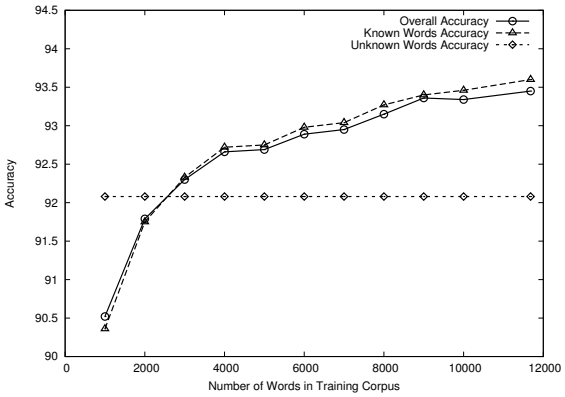

Figure 2: POS Learning Curve

the best of our knowledge no comparable results have been reported so far for Hindi.

From Table 1, we can see that the disambiguation module brings up the accuracy of simple lexicon lookup based approach by around 25% (LLBD). The overall average accuracy is also brought up by around 20% by augmenting the morphology-driven (MD) tagger by a disambiguation module; hence justifying our belief that a disambiguation module over a morphology driven approach yields better results.

One interesting observation is the performance of the tagger on individual POS categories. Figure 3 shows clearly that the per POS accuracies of the LB tagger highly exceeds those of the MD and BL tagger for most categories. This means that the disambiguation module correctly disambiguates and correctly identifies the unknown words too. The accuracy on unknown words, as earlier shown in Figure 2, is very high at 92.08%. The percentage of unknown words in the test corpora is 0.013. It seems independent of the size of training corpus because the corpora is unbalanced having most of the unknowns as proper nouns. The rules are formed on this bias, and hence the application of these rules assigns PPN tag to an unknown which is mostly the case.

From Figure 3 again we see that the accuracy on some categories remains very low even after disambiguation. This calls for some detailed failure analysis. By looking at the categories having low accuracy, such as pronoun, intensifier, demonstratives and verb copula, we find that all of them are highly ambiguous and, almost invariably, very rare in the corpus. Also, most of them are hard to disambiguate without any semantic information.

7 Conclusions & Future Work

We have described in this paper a POS tagger for Hindi which can overcome the handicap of annotated corpora scarcity by exploiting the rich morphology of the language and the relatively rigid word-order within a VG. The whole work was driven by hunting down the factors that lower the accuracy of *Verbs* and weeding them out. A detailed study of accuracy distribution across the POS tags pointed out the cases calling for elaborate disambiguation rules. A major strength of the work is the learning of disambiguation rules, which otherwise would have been hand-coded, thus demanding exhaustive analysis of language phenomena. Attaining an accuracy of close to 94%, from a corpora of just about 15,562 words lends credence to the belief that "*morphological richness can offset resource scarcity*". The work could lead such efforts of POS tag building for all those languages which have rich morphology, but cannot afford to invest a lot in creating large annotated corpora.

Several interesting future directions suggest themselves. It will be worthwhile to investigate a statistical approach like Conditional Random Fields in which the feature functions would be constructed from morphology. The next logical step from the POS tagger is a chunker for Hindi. In fact a start on this has already been made through VG identification.

References

A. Ratnaparakhi. 1996. *A Maximum Entropy Part-Of-Speech Tagger.* EMNLP 1996

A. Bharati, V. Chaitanya, R. Sangal 1995. *Natural Language Processing : A Paninian Perspective* . Prentice Hall India.

A. Kuba, A. Hcza, J. Csirik 2004. *POS Tagging of Hungarian with Combined Statistical and Rule-Based Methods.* TSD 2004

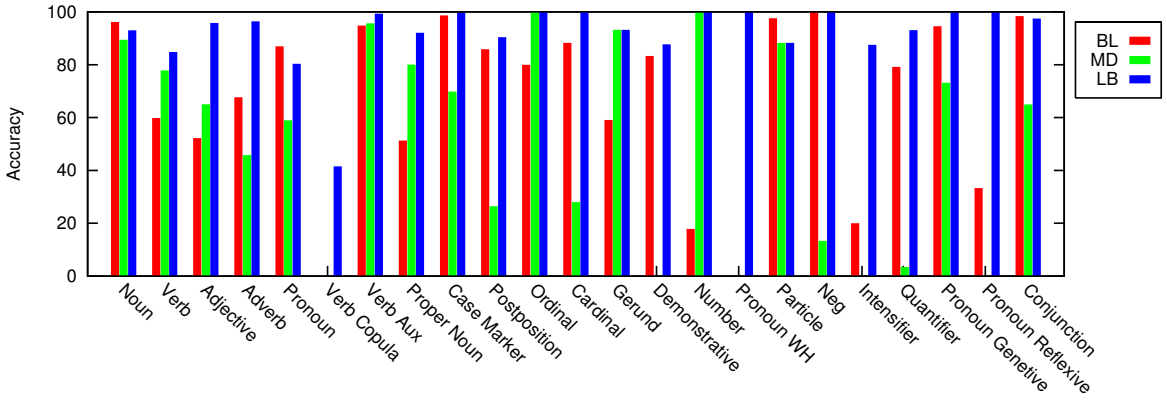

Figure 3: Per-POS Accuracy Distribution

B. Megyesi. 1999. *Improving Brill's POS tagger for an agglutinative language.* Joint Sigdat Conference EMNLP/VLC 1999.

C. D. Manning and H. Schutze. 2002. *Foundations of Statistical Natural Language Processing,* MIT Press 2002.

D. Cutting et al. 1992. *A practical part-of-speech tagger.* In Proc. of the Third Conf. on Applied Natural Language Processing. ACL 1992.

E. Brill. 1995. *Transformation-Based Error Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging* . Computational Linguistics 21(94): 543-566. 1995.

E. Black et al. 1992. *Decision tree models applied to the labeling of text with parts-of-speech.* In Darpa Workshop on Speech and Natural Language 1992.

G. Leech, R. Garside and M. Bryant. 1992. *Automatic POS-Tagging of the corpus.* BNC2 POS-tagging Manual.

G. Orphanos, D. Kalles, A. Papagelis, D. Christodoulakis. 1999 *Decision trees and NLP: A Case Study in POS Tagging* . In proceedings of ACAI 1999.

H. Schmid 1994 *Part-of-Speech Tagging with Neural Networks* . In proceedings of COLING 1994.

J. Hajic, P. Krbec, P. Kveton, K. Oliva, V. Petkevic 2001 *A Case Study in Czech Tagging.* In Proceedings of the 39th Annual Meeting of the ACL 2001

K. Uchimoto, S. Sekine, 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 2001

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 1994

M. Shrivastava, N. Agrawal, S. Singh, P. Bhattacharya. 2005. *Harnessing Morphological Analysis in POS Tagging Task.* In Proceedings of the ICON 2005

P. R. Ray , V. Harish, A. Basu and S. Sarkar 2003. *Part of Speech Tagging and Local Word Grouping Techniques for Natural Language Parsing in Hindi*. In Proceedings of ICON 2003

P. Clark and T. Niblett 1989. *The CN2 Induction Algorithm.* Journal of Machine Learning, vol(3), pages 261-283, 1989

R. Garside, N. Smith 1997. *A hybrid grammatical tagger: CLAWS4* . In R. Garside, G. Leech, A. McEnery (eds.) Corpus annotation: Linguistic information from computer text corpora. Longman. Pp. 102-121.

C. Samuelsson and A. Voutilainen 1997. *Comparing a Linguistic and a Stochastic Tagger.* In Procs. Joint 35th Annual Meeting of the ACL and 8th Conference of the European Chapter of the ACL 1997.

Y. Tlili-Guiassa 2006. *Hybrid Method for Tagging Arabic Text*. Journal of Computer Science 2 (3): 245-248, 2006