Statistical Morphological Analyzer for Hindi

Deepak Kumar Malladi and **Prashanth Mannem**
Language Technologies Research Center
International Institute of Information Technology
Hyderabad, AP, India - 500032
{deepak.malladi, prashanth}@research.iiit.ac.in

Abstract

Morphology is the study of internal structure of words and is an essential early step in many NLP applications such as parsing and machine translation. Researchers working in Hindi NLP have either used the widely popular paradigm based analyzer (PBA) or extensions of it. In this work, we undertook a comprehensive evaluation of PBA using the data from the Hindi Treebank (HTB) and presented a new morphological analyzer trained on the HTB. Our morphological analyzer has better coverage and accuracy when compared to the existing analyzers for Hindi. An oracle system that takes the best values from the PBA's output achieves only 63.41% for lemma, gender, number, person and case. Our statistical analyzer has an accuracy of 84.16% for these morphological attributes when evaluated on the test section of the Hindi Treebank.

1 Introduction

Morphological analysis is the task of analyzing the structure of morphemes in a word and is generally a prelude to further complex NLP tasks such as parsing, machine translation, semantic analysis etc. These tasks need an analysis of the words in the sentence in terms of lemma, affixes, parts of speech (POS) etc.

Hindi is a morphologically rich language with a relatively free word order. Previous efforts in Hindi morphological analysis concentrated on building rule-based systems that give all the possible analyses for a word form irrespective of its context in the sentence. The paradigm based analyzer (PBA) by Bharati et al. (1995) is one of the most widely used applications among researchers in the Indian NLP community. In paradigm based analysis, words are grouped into a set of paradigms depending on the inflections they take. Each paradigm has a set of add-delete rules to account for its inflections and words belonging to a paradigm take the same inflectional forms. Given a word, the PBA identifies the *lemma*, *coarse POS tag*, *gender*, *number*, *person*, *case marker*, *vibhakti*[1] and *TAM* (tense, aspect, modality). Being a rule-based system, the PBA takes a word as input and gives all the possible analyses as output. (Table 1 presents an example). It doesn't pick the correct analysis for a word in its sentential context.

Goyal and Lehal's analyser (2008), which is a re-implementation of the PBA with few extensions, has not done any comparative evaluation. Kanuparthi et al. (2012) built a derivational morphological analyzer for Hindi by introducing a layer over the PBA. It identifies 22 derivational suffixes which helps in providing derivational analysis for the word whose suffix matches with one of these 22 suffixes.

The large scale machine translation projects[2] that are currently under way in India use shallow parser built on PBA and an automatic POS tagger. The shallow parser prunes the morphological analyses from PBA to select the correct one using the POS tags from the tagger. Since it is based on PBA, it suffers from similar coverage issues for out of vocabulary (OOV) words.

The PBA, developed in 1995, has a limited vocabulary and has received only minor upgrades since then. Out of 17,666 unique words in the Hindi Treebank (HTB) released during the 2012 Hindi Parsing Shared Task (Sharma et al., 2012), the PBA does not have entries for 5,581 words (31.6%).

NLP for Hindi has suffered due to the lack of a

[1] Vibhakti is a Sanskrit grammatical term that encompasses post-positionals and case endings for nouns, as well as inflection and auxiliaries for verbs (Pedersen et al., 2004).

[2] http://sampark.iiit.ac.in/

International Joint Conference on Natural Language Processing, pages 1007–1011,
Nagoya, Japan, 14-18 October 2013.

	L ↓	G ↓	N ↓	P ↓	C ↓	T/V ↓
xeSa (*country*)	xeSa	m	sg	3	d	0
	xeSa	m	pl	3	d	0
	xeSa	m	sg	3	o	0
cAhie (*want*)	cAha	any	sg	2h	_	ie
	cAha	any	pl	2h	_	eM

L-lemma, G-gender, N-number, P-person
C-case, T/V-TAM or Vibhakti

Table 1: Multiple analyses given by the PBA for the words xeSa and cAhie

high coverage automatic morphological analyzer. For example, the 2012 Hindi Parsing Shared Task (Sharma et al., 2012) held with COLING-2012 workshop had a gold-standard input track and an automatic input track, where the former had gold-standard morphological analysis, POS tags and chunks of a sentence as input and the automatic track had only the sentence along with automatic POS tags as input. The morphological information which is crucial for Hindi parsing was missing in the automatic track as the existing analyzer had limited coverage.

In this work, we present a statistical morphological analyzer for Hindi trained on HTB and compare it with PBA. The analyzer predicts the *lemma*, *gender*, *number*, *person* (GNP) and *case marker* for all the words in a given sentence by training separate models on the HTB for each of them. Other grammatical features such as TAM (tense, aspect, modality) and *vibhakti* are predicted using heuristics on fine grained POS tags of the input sentence. Our system has significantly better accuracy than analyzers based on PBA and is robust enough to produce analyses for OOV words.

2 Statistical Morphological Analyzer (SMA)

The output of a morphological analyzer depends on the language that it is developed for. Analyzers for English (Goldsmith, 2000) predict just the lemmas and affixes mainly because of its restricted agreement based on semantic features such as animacy and *natural* gender. But in Hindi, agreement depends on *lexical* features such as *grammatical* gender, number, person and case. Hence, it is crucial that Hindi analyzers predict these along with TAM and vibhakti which have been found to be useful for syntactic parsing (Ambati et al., 2010; Bharati et al., 2009a).

MorphFeature	Values
Gender	masculine, feminine, any, none
Number	singular, plural, any, none
Person	1, 1h, 2, 2h, 3, 3h, any, none
CaseMarker	direct, oblique, any, none

Table 2: Morph features and the values they take

Hindi has syntactic agreement (of GNP and case) of two kinds: modifier-head agreement and noun-verb agreement. Modifiers, including determiners, agree with their head noun in gender, number and case, and finite verbs agree with some noun in the sentence in gender, number and person (Kachru, 2006). Therefore, apart from lemma and POS tags, providing gender, number and person is also crucial for syntactic parsing.[3]

With the existing morph analyzer (PBA) performing poorly on OOV (unknown to PBA) words and the availability of an annotated treebank, we set out to build a high-coverage automatic Hindi morph analyzer by learning each of the seven morphological attributes separately from the Hindi Treebank. During this process, it was realized that vibhakti and TAM can be better predicted using heuristics on fine-grained POS tags than by training on the HTB.

In the rest of the section, we discuss the methods to predict each of the seven morphological attributes. Table 2 lists the values that each of the morph attributes take in HTB. The HTB consists of 15,102 sentences (334,287 words) annotated with morphological features, POS tags, chunks and dependency relations. In this work, we only use morph and POS information.

2.1 Lemma prediction

The PBA uses a large vocabulary along with paradigm tables consisting of add-delete rules to find the lemma of a given word. All possible add-delete rules are applied on a given word form and the resulting lemma is checked against the vocabulary to find if it is right or not. If no such lemma exists (for OOV words), it returns the word itself as the lemma.

While the gender, number and person of a word form varies according to the context (due to syntactic agreement with head words), there are very

[3]While nouns, pronouns and adjectives have both GNP and case associated with them, verbs only have GNP. TAM is valid only for verbs and vibhakti (post-position) is only associated with nouns and pronouns.

Analysis	Test Data - Overall(%)				Test Data - OOV of SMA(%)			
	Baseline	F-PBA	O-PBA	SMA	Baseline	F-PBA	O-PBA	SMA
L	71.12	83.10	86.69	95.70	78.10	82.08	82.48	85.82
G	37.43	72.98	79.59	95.43	60.22	43.07	44.06	79.09
N	52.87	72.22	80.50	94.90	69.60	44.53	47.56	89.12
P	45.59	74.33	84.13	95.77	78.30	52.51	53.89	94.39
C	29.31	58.24	81.20	94.62	43.60	31.40	47.36	87.40
V/T	65.40	53.05	59.65	97.04	58.31	33.58	34.56	96.04
L+C	16.46	48.84	72.06	90.67	32.52	28.50	44.66	75.33
L+V/T	54.78	44.57	51.71	92.93	53.56	31.73	32.72	82.65
G+N+P	23.05	61.10	73.81	89.42	47.49	35.75	39.58	71.31
G+N+P+C	9.72	45.73	70.87	85.56	21.04	20.91	35.95	64.64
L+G+N+P	20.27	53.29	66.28	85.88	44.72	34.63	38.46	62.34
L+G+N+P+C	8.57	38.25	63.41	**82.16**	19.33	19.92	34.89	**56.66**
L+G+N+P+C+V/T	1.25	32.53	42.80	**80.11**	4.02	14.51	18.67	**54.35**

L-lemma, G-gender, N-number, P-person, C-case, V/T-Vibhakti/TAM

Table 3: Accuracies of SMA compared with F-PBA, O-PBA and baseline systems.

few cases where a word form can have more than one lemma in a context. This makes lemma simpler to predict among the morphological features, provided there is access to a dictionary of all the word forms along with their lemmas. Unfortunately, such a large lemma dictionary doesn't exist.

In this work, we perceived lemma prediction from a machine translation perspective, with the characters in the input word form treated as the source sentence and those in the lemma as the target. The strings on source and target side are split into sequences of characters separated by space. The phrase based model (Koehn et al., 2007) in Moses is trained on the parallel data created from the training part of HTB. The translation model accounts for the changes in the affixes (sequence of characters) from word form to lemma whereas the language model accounts for which affixes go with which stems. In this perspective, the standard MT experiment of switching source and target to attain better accuracy would not apply since it is unreasonable to predict the word form from the lemma without taking the context into account.

2.2 Gender, Number, Person and Case Prediction

Unlike lemma prediction, we use a liblinear classifier (Fan et al., 2008) to build linear SVM classification models for GNP and case prediction.

Though knowing the syntactic head of a word helps in enforcing agreement (and thereby accurately predicting the correct GNP), parsing is usually a higher level task and is not performed before morphological analysis. Hence, certain cases of GNP prediction are similar in nature to the standard chicken and egg problem.

The following features were tried out in building the models for gender, number, person and case prediction:

- Word
- Lexical category
- Last 3 characters
- Last 4 characters
- Next word
- Previous word
- Lemma
- Word Length
- Character N-grams of the word

2.3 Vibhakti and TAM

Vibhakti and TAM are helpful in identifying the *karaka*[4] dependency labels in HTB. While nouns and pronouns take vibhakti, verbs inflect for TAM. Both TAM and vibhakti occur immediately after the words in their respective word classes.

Instead of building statistical models for vibhakti and TAM prediction, we built a system that uses heuristics on POS tag sequences to predict the correct value. The POS tags of words following nouns, pronouns and verbs give an indication as to what the vibhakti/TAM are. Words with PSP (postposition) and NST (noun with spatial and temporal properties) tags are generally considered as the vibhakti for the preceding nouns and

[4]karakas are syntactico-semantic relations which are employed in Paninian framework (Begum et al., 2008; Bharati et al., 2009b)

Data	#Sentences	#Words
Training	12,041	268,096
Development	1,233	26,416
Test	1,828	39,775

Table 4: HTB statistics

pronouns. A postposition in HTB is annotated as PSP only if it is written separately (usane/PRP vs usa/PRP ne/PSP). For cases where the postposition is not written separately we rely on the treebank data to get the suffix. Similarly, words with VAUX tag form the TAM for the immediately preceding verb.

The PBA takes individual words as input and hence does not output the entire vibhakti or TAM of the word in the sentence. It only identifies these values for those words which have the information within the word form (e.g. usakA *he+Oblique*, kiyA *do+PAST*).

In the sentence,

rAma/NNP kA/PSP kiwAba/NN cori/NN ho/VM sakawA/VAUX hE/VAUX,

PBA identifies rAma's vibhakti as *0* and ho's TAM as *0*. Whereas in HTB, vibhakti and TAM of rAma and ho are annotated as *0_kA* and *0_saka+wA_hE* respectively. Our approach determines this information precisely.

3 Experiments and Results

The Hindi treebank released as part of the 2012 Hindi Parsing Shared Task is used to evaluate our models. All the models are tuned on development data and evaluated on test data. Table 4 shows the word counts of training, development and test sections of HTB.

Our approach to Hindi morphological analysis is based on handling each of the seven attributes (*lemma, gender, number, person, case, vibhakti* and *TAM*) separately. However, evaluation is performed on individual attributes as well as on the combined output. The models are compared with a baseline system and two versions of the PBA wherever relevant. The *baseline* system takes the word form itself as the lemma and selects the most frequent value for the rest of the attributes.

Since PBA is a rule-based analyzer which gives more than one analysis for words, we use two versions of it for comparison. The first system is the oracle PBA (referred further as O-PBA) which uses an oracle to pick the *best* analysis from the list of all analyses given by the PBA. The second version of the PBA (F-PBA) picks the *first* analysis from the output as the correct analysis.

Table 3 presents the accuracies of four systems (baseline, F-PBA, O-PBA and SMA) in predicting the morphological attributes of all the words in the HTB's test data and also for OOV words of SMA (i.e. words that occur in the test section but not in training section of HTB)[5]. The accuracies are the percentages of words in the data with the correct analysis. It may be noted that our system (SMA) performs significantly better than the best analyses of PBA and the baseline system in all the experiments conducted.

The existing Hindi POS tagger[6] was found to be 95% accurate when evaluated on the entire HTB data. We got similar results when we had run the entire set of experiments using these automatic POS tags.

4 Conclusion and Future work

In conclusion, our paper presented a robust state-of-the-art statistical morphological analyzer for Hindi which outperforms previous analyzers by a considerable margin. A comprehensive evaluation was carried out for our system by comparing it with the existing analyzers. The analyzer we have developed achieved an accuracy of 82.03% for lemma, gender, number, person, case, vibhakti and TAM. Being a statistical model, it even analyzes OOV words thereby extending the coverage of the analyzer. We also evaluated the effect of morphological features (predicted by our system) on dependency parsing and found them to improve the parsing accuracy.

The agreement phenomenon in Hindi provides challenges in predicting gender, number and person of words in their sentential context. These can be better predicted if dependency relations are given as input. However, the standard natural language analysis pipeline forbids using parse information during morphological analysis.

This provides an opportunity to explore joint modelling of morphological analysis and syntactic parsing for Hindi. We plan to experiment this as part of our future work.

[5]OOV words for SMA need not be *out of vocabulary* for PBA's dictionaries.

[6]ilmt.iiit.ac.in

References

Bharat Ram Ambati, Samar Husain, Joakim Nivre, and Rajeev Sangal. 2010. On the role of morphosyntactic features in hindi dependency parsing. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 94–102. Association for Computational Linguistics.

Rafiya Begum, Samar Husain, Arun Dhwaj, Dipti Misra Sharma, Lakshmi Bai, and Rajeev Sangal. 2008. Dependency annotation scheme for indian languages. In *Proceedings of IJCNLP*.

Akshar Bharati, Vineet Chaitanya, Rajeev Sangal, and KV Ramakrishnamacharyulu. 1995. *Natural language processing: A Paninian perspective*. Prentice-Hall of India New Delhi.

Akshar Bharati, Samar Husain, Meher Vijay, Kalyan Deepak, Dipti Misra Sharma, and Rajeev Sangal. 2009a. Constraint based hybrid approach to parsing indian languages. *Proc of PACLIC 23. Hong Kong*.

Akshara Bharati, Dipti Misra Sharma, Samar Husain, Lakshmi Bai, Rafiya Begam, and Rajeev Sangal. 2009b. Anncorra: Treebanks for indian languages, guidelines for annotating hindi treebank.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

John Goldsmith. 2000. Linguistica: An automatic morphological analyzer. In *Proceedings of 36th meeting of the Chicago Linguistic Society*.

Vishal Goyal and G. Singh Lehal. 2008. Hindi morphological analyzer and generator. In *Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on*, pages 1156–1159. IEEE.

Yamuna Kachru. 2006. *Hindi*, volume 12. John Benjamins Publishing Company.

Nikhil Kanuparthi, Abhilash Inumella, and Dipti Misra Sharma. 2012. Hindi derivational morphological analyzer. In *Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology*, pages 10–16. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics.

Mark Pedersen, Domenyk Eades, Samir K Amin, and Lakshmi Prakash. 2004. Relative clauses in hindi and arabic: A paninian dependency grammar analysis. *COLING 2004 Recent Advances in Dependency Grammar*, pages 9–16.

Dipti Misra Sharma, Prashanth Mannem, Joseph Van-Genabith, Sobha Lalitha Devi, Radhika Mamidi, and Ranjani Parthasarathi, editors. 2012. *Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages*. Mumbai, India, December.