Context Based Statistical Morphological Analyzer and its Effect on Hindi Dependency Parsing

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

This paper revisits the work of (Malladi and Mannem, 2013) which focused on building a *Statistical Morphological Analyzer* (SMA) for Hindi and compares the performance of SMA with other existing statistical analyzer, *Morfette*. We shall evaluate SMA in various experiment scenarios and look at how it performs for unseen words. The later part of the paper presents the effect of the predicted morph features on dependency parsing and extends the work to other morphologically rich languages: Hindi and Telugu, without any language-specific engineering.

1 Introduction

Hindi is one of the widely spoken language in the world with more than 250 million native speakers[1]. Language technologies could play a major role in removing the digital divide that exists between speakers of various languages. Hindi, being a morphologically rich language with a relatively free word order (Mor-FOW), poses a variety of challenges for NLP that may not be encountered when working on English.

Morphological analysis is the task of analyzing the structure of morphemes in a word and is generally a prelude to further complex 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.

[1] http://www.ethnologue.com/statistics/size

NLP for Hindi has suffered due to the lack of a 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. Parsing accuracies of gold-standard input track were significantly higher than that of the other track. But in the real scenario NLP applications, gold information is not provided. Even Ambati et al. (2010b) and Bharati et al. (2009a) have exploited the role of morpho-syntactic features in Hindi dependency parsing. Hence we need a high coverage and accurate morphological analyzer.

2 Related work

Previous efforts on 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 based 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

Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically Rich Languages, pages 119–128,
Seattle, Washington, USA, 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

word, the PBA identifies the *lemma*, *coarse POS tag*, *gender*, *number*, *person*, *case marker*, *vibhakti*[2] 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[3] 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%).

Apart from the traditional rule-based approaches, Morfette (Chrupała et al., 2008) is a modular, data-driven, probabilistic system which learns to perform joint morphological tagging and lemmatization from morphologically annotated corpora. The system is composed of two learning modules, one for morphological tagging and one for lemmati- zation, and one decoding module which searches for the best sequence of pairs of morphological tags and lemmas for an input sequence of wordforms.

Malladi and Mannem (2013) have build a *Statistical Morphological Analyzer* (SMA) with minimal set of features but they haven't compared their system with Morfette. In our work we shall discuss in detail about SMA with more concentration on evaluating the system in various scenarios and shall extend the approach to other morphologically rich languages. Later we evaluate the effect of the predicted morph features (by SMA) on Hindi dependency parsing.

[2]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).

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

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

Table 2: HTB statistics

3 Hindi Dependency Treebank (HTB)

A multi layered and multi representational treebank for Hindi is developed by annotating with morpho-syntactic (morphological analyses, POS tags, chunk) and syntacto-semantic (dependency relations labeled in the computational paninian framework) information. A part of the HTB (constituting of 15,102 sentences) was released for Hindi Parsing Shared Task. Table 2 shows the word counts of training, development and test sections of HTB.

With the existing morph analyzer (PBA) performing poorly on OOV words and the availability of an annotated treebank, Malladi and Mannem (2013) 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 used by SMA to predict each of the seven mor-

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 3: Morph features and the values they take

source	target	gloss
k i y A	k a r a	*do*
l a d a k e	l a d a k A	*boy*
l a d a k I	l a d a k I	*girl*
l a d a k I y A M	l a d a k I	*girl*

Table 4: Sample parallel corpus for lemma prediction

phological attributes and their effect on Hindi dependency parsing. Table 3 lists the values that each of the morph attributes take in HTB.

4 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., 2010b; Bharati et al., 2009a).

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.[4]

[4]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.

4.1 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 few cases where a word form can have more than one lemma in a context. For example, vaha can either be masculine or feminine depending on the form that the verb takes. It is feminine in vaha Gara gayI (*she went home*) and masculine in vaha Gara gayA (*he went home*). The lemma for vaha can only be vaha irrespective of the context and also the lemma for gayI and gayA is jA. 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. There are 15,752 word types in training, 4,292 word types in development and 5,536 word types in test sections of HTB respectively. Among these 18.5% of the types in development and 20.2% in test data are unseen in training data.

SMA analyzer perceives 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, as shown in Table 4. 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.

Apart from the above mentioned approach, we apply a heuristic on top of SMA, wherein proper nouns

Gender	Word	Gloss
masculine	cAvala, paMKA	*rice*, *fan*
feminine	rela, xAla	*train*, *pulse*
any	jA	*go*
none	karIba	*near*

Table 5: Gender value examples

Number	Word	Gloss
singular	ladZake	*boy-Sg-Oblique*
plural	ladZake	*boy-Pl-Direct*
any	banA	*make*
none	karIba	*near*

Table 6: Number value examples

(NNP) take the word form itself as the lemma.

4.2 Gender, Number, Person and Case Prediction

Unlike lemma prediction, SMA uses SVM (support vector machine) machine learning algorithm to predict GNP and case.

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.

4.2.1 Gender

Gender prediction is tricky in Hindi as even native speakers tend to make errors while annotating. Gender prediction in English is easy when compared to Hindi since gender in English is inferred based on the biological characteristics the word is referring to. For example, Train has neuter gender in English whereas in Hindi, it exhibits feminine characteristics. A dictionary of word-gender information may usually suffice for gender prediction in English but in Hindi it isn't the case as gender could vary based on its agreement with verb/modifier. The values that gender can take for a word in a given context are *masculine*(*m*), *feminine*(*f*), *any* (either *m* or *f*) or *none* (neither *m* nor *f*). Table 5 gives example for each gender value.

Nouns inherently carry gender information. Pronouns (of genitive form), adjectives and verbs inflect according to the gender of the noun they refer to.

Case	Word	Gloss
direct	ladZake	*boy-Pl*
oblique	ladZake	*boy-sg*
any	bAraha	*twelve (cardinals)*
none	kaha	*say*

Table 7: Case value examples

4.2.2 Number

Every noun belongs to a unique number class. Noun modifiers and verbs have different forms for each number class and inflect accordingly to match the grammatical number of the nouns to which they refer.

Number takes the values *singular* (*sg*), *plural* (*pl*), *any* (either *sg* or *pl*) and *none* (neither *sg* nor *pl*). Table 6 lists examples for each of the values. In it, ladZake takes the grammatical number *sg* (in *direct* case) or *pl* (in *oblique* case) depending on the context in which it occurs. It may be noted that since PBA does not consider the word's context, it outputs both the values and leaves the disambiguation to the subsequent stages.

4.2.3 Person

Apart from *first*, *second* and *third* persons, Hindi also has the honorific forms, resulting in *1h*, *2h* and *3h*. Postpositions do not have person information, hence *none* is also a possible value. Apart from the above mentioned grammatical person values, *any* is also a feasible value.

4.2.4 Case Marker

Case markers in Hindi (*direct* and *oblique*) are attributed to nouns and pronouns. Table 7 lists few examples.

Words which inflect for gender, number, person and case primarily undergo affixation at the end.

Features for GNP & Case Marker

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

- Word level features
  - Word

  - Last 2 characters
  - Last 3 characters
  - Last 4 characters
  - Character N-grams of the word
  - Lemma
  - Word Length
- Sentence level features
  - Lexical category[5]
  - Next word
  - Previous word

Combinations of these features have been tried out to build the SVM models for GNP and case. For each of these tasks, feature tuning was done separately. In Malladi and Mannem (2013), a linear SVM classification (Fan et al., 2008) is used to build statistical models for GNP and case but we found that with RBF kernel (non-linear SVM)[6] we achieve better accuracies. Furthermore, the parameters (C, γ) of the RBF kernel are learned using grid search technique.

4.3 Vibhakti and TAM

Vibhakti and TAM are helpful in identifying the *karaka*[7] 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, SMA 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 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 SMA relies 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. SMA determines this information precisely and Morfette which can predict other morph features, is not capable of predicting TAM and Vibhakti as these features are specific to Indian languages.

5 Evaluation Systems

SMA is compared with a baseline system, Morfette 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.

Morfette can perdict lemma, gender, number, person and case attributes but it cannot predict TAM and Vibhakti as they do not have a definite set of predefined values unlike other morphological attributes.

6 Experiments and Results

SMA 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

[5]POS is considered as a sentence level feature since tagging models use the word ngrams to predict the POS category

[6]LIBSVM tool is used to build non-linear SVM models for our experiments (Chang and Lin, 2011).

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

Analysis	Test Data - Overall(%)					Test Data - OOV of SMA(%)				
	Baseline	F-PBA	O-PBA	Morfette	SMA	Baseline	F-PBA	O-PBA	Morfette	SMA
L	71.12	83.10	86.69	94.14	95.84	78.10	82.08	82.48	90.30	89.51
G	37.43	72.98	79.59	95.05	96.19	60.22	43.07	44.06	72.03	82.65
N	52.87	72.22	80.50	94.09	95.37	69.60	44.53	47.56	84.89	90.44
P	45.59	74.33	84.13	94.88	96.38	78.30	52.51	53.89	84.76	94.85
C	29.31	58.24	81.20	93.91	95.32	43.60	31.40	47.36	80.21	88.52
V/T	65.40	53.05	59.65	NA	97.04	58.31	33.58	34.56	NA	96.04
L+C	16.46	48.84	72.06	88.56	91.39	32.52	28.50	44.66	72.89	79.09
L+V/T	54.78	44.57	51.71	NA	93.06	53.56	31.73	32.72	NA	86.41
G+N+P	23.05	61.10	73.81	88.36	91.11	47.49	35.75	39.58	62.33	76.52
G+N+P+C	9.72	45.73	70.87	84.43	87.78	21.04	20.91	35.95	55.74	69.99
L+G+N+P	20.27	53.29	66.28	83.44	87.51	44.72	34.63	38.46	57.85	69.13
L+G+N+P+C	8.57	38.25	63.41	79.73	**84.25**	19.33	19.92	34.89	51.52	**63.06**
L+G+N+P+C+V/T	1.25	32.53	42.80	NA	**82.12**	4.02	14.51	18.67	NA	**60.07**

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

Table 8: Accuracies of SMA compared with F-PBA, O-PBA and baseline systems.

on individual attributes as well as on the combined output.

SMA builds models for lemma, gender, number, person and case prediction trained on the training data of the HTB. All the models are tuned on development data and evaluated on test data of the HTB.

Table 8 presents the accuracies of five systems (baseline, F-PBA, O-PBA, Morfette 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)[8]. The accuracies are the percentages of words in the data with the correct analysis. It may be noted that SMA performs significantly better than the best analyses of PBA and the baseline system in all the experiments conducted. As far as Morfette is concerned, it performs on par with SMA in terms of overall accuracy but for OOV words, except for lemma prediction, SMA outperforms Morfette by significant margin.

Table 13 lists the accuracies of lemma, gender, number, person and case for the most frequently occurring POS tags. Table 12 reports the same for OOV words. The number of OOV words in postposition and pronoun categories is quite less and hence have not been included in the table.

[8]OOV words for SMA need not be *out of vocabulary* for PBA's dictionaries. Table 8 lists accuracies for OOV words of SMA. We shall also report accuracies for OOV words of PBA in the later part of the paper (Table 11).

Metric	Exp-1[a]	Exp-2[b]	Exp-3[c]
LAS	87.75	89.41	89.82
UAS	94.41	94.50	94.81
LA	89.89	91.67	91.96

Table 9: MALT Parser's accuracies on HTB test data. Unlabeled Attachment Score (UAS) is the percentage of words with correct heads. Labeled Accuracy (LA) is the percentage of words with correct dependency labels. Labeled Attachment Score (LAS) is the percentage of words with both correct heads and labels.

[a]Exp-1: Without morph features
[b]Exp-2: With morph features predicted by SMA
[c]Exp-3: With gold morph features (as annotated in HTB)

Hindi derivational morph analyzer (Kanuparthi et al., 2012) and the morph analyzer developed by Punjab University (Goyal and Lehal, 2008) do not add much to PBA accuracy since they are developed with PBA as the base. Out of 334,287 words in HTB, the derivational morph analyzer identified only 9,580 derivational variants. For the remaining words, it gives similar analysis as PBA.

6.1 Lemma

The evaluation metric for lemma's model is *accuracy*, which is the percentage of predicted lemmas

that are correct. The phrase based translation system used to predict lemmas achieved an accuracy of 95.84% compared to O-PBA's 86.69%. For OOV words, the PBA outputs the word itself as the lemma whereas the translation-based lemma model is robust enough to give the analysis.

The translation-based lemma model and O-PBA report accuracies of 89.51% and 82.48% respectively for OOV words of SMA. In terms of both overall and OOV accuracies, translation-based model outperforms PBA. Though SMA performs better than Morfette in terms of overall accuracy, but for OOV accuracy Morfette narrowly outperforms SMA.

The postposition accuracy is significantly worse than the overall accuracy. This is because the confusion is high among postpositions in HTB. For example, out of 14,818 occurrences of `ke`, it takes the lemma `kA` in 7,763 instances and `ke` in 7,022 cases. This could be the result of an inconsistency in the annotation process of HTB. The accuracies for verbs are low (when compared to Nouns, Adjectives) as well mainly because verbs in Hindi take more inflections than the rest. The accuracy for verbs is even lower for OOV words (69.23% in Table 12).

6.2 Gender, Number, Person and Case

The accuracies of gender, number, person and case hover around 95% but the combined (G+N+P) accuracy drops to 91.11%. This figure is important if one wants to enforce agreement in parsing.

The OOV accuracy for person is close to overall accuracy as most of the OOV words belong to the 3rd person category. It is not the same case for gender and number. Gender particularly suffers a significant drop of 14% for OOV words confirming the theory that gender prediction is a difficult problem without knowing the semantics of the word.

The number and person accuracies for verbs are consistently low for OOV words as well as for seen words. This could be because SMA doesn't handle long distance agreement during GNP prediction.

Until now, we reported accuracies for OOV words of SMA. Table 11 lists accuracies for OOV words of the PBA (i.e. words which are not analyzed by the PBA) in the test section of HTB. SMA clearly outperforms baseline system and also performs better than F-PBA and O-PBA as they do not give any analyses.

Analysis	Accuracy	OOV Accuracy
Gender	95.74	80.08
Number	95.29	89.71
Person	96.12	94.06
Case	95.16	88.32
G+N+P	90.92	74.14
G+N+P+C	87.72	68.47

Table 10: Joint Model for Gender, Number, Person, Case

In a nutshell, we have evaluated SMA for OOV words of the PBA as well as for OOV words of SMA. In both the cases, SMA performed better than other systems. We shall evaluate SMA in a challenging scenario wherein *training* data consists of the words from the HTB which are analyzed by the PBA and *test* data consists of the remaining unanalyzed words by the PBA. Thereby, the entire test data contains only *out of vocabulary* instances for both SMA and PBA. Table 14 presents the results of this new evaluation. The results are almost similar with that of OOV results shown in Table 8 except for *Person*. The reason behind that could be, in the training data there are only 0.1% instances of *3h* class but in test data their presence is quite significant (approximately 10%). The training instances for *3h* class were not sufficient for the model to learn and hence very few of these instances were identified correctly. This explains the drop in *Person* accuracy for this experiment scenario.

It may be noted that, we have used gold POS tags for all our experiments related to GNP and case prediction. There are numerous efforts on building POS taggers for Hindi. The ILMT pos tagger[9] is 96.5% accurate on the test data of the HTB. Table 15 reports the accuracies of gender, number, person and case using the automatic POS tags predicted by the ILMT tagger. The results are similar to that of the experiments conducted with gold POS tags.

Malladi and Mannem (2013) have build separate models for gender, number, person and case. Table 10 reports the results of *Joint Model* for these morph attributes. In terms of accuracy, Joint Model is as efficient as individual models.

[9] http://ilmt.iiit.ac.in/

Analysis	Baseline	SMA
Lemma	65.40	95.96
Gender	57.09	95.93
Number	76.79	95.17
Person	65.76	96.42
Case	46.39	95.17

Table 11: Accuracy for OOV words of PBA

Analysis	Noun	Verb	Adjective
Lemma	92.18	69.23	88.35
Gender	80.49	86.15	92.23
Number	92.35	76.92	87.38
Person	96.64	75.38	100.00
Case	88.81	98.46	70.87

Table 12: OOV accuracies for words (by POS tags)

6.3 TAM and Vibhakti

The proposed heuristics for Vibhakti and TAM prediction gave accuracy of 97.04% on test data set of HTB. On the entire HTB data, SMA achieved accuracy of 98.88%. O-PBA gave accuracy of 59.65% for TAM and Vibhakti prediction on test part of HTB. The reason behind low performance of O-PBA is that it identifies the TAM and vibhakti values for each word separately and doesn't consider the neighbouring words in the sentence.

7 Effect on Parsing

The effect of morphological features on parsing is well documented (Ambati et al., 2010a). Previous works used gold morphological analysis to prove their point. In this work, we also evaluated the effect of *automatic* morph features (predicted by SMA) on dependency parsing. MALT parser was trained on gold-standard POS tagged HTB data with and with out morph features. Table 9 lists the evaluation scores for these settings. While the unlabeled attachment score (UAS) does not show significant improvement, the labeled attachment score (LAS) and label accuracy (LA) have increased significantly. Ambati et al. (2010a) also reported similar results with *gold-standard* morph features. Lemma, case, vibhakti and TAM features contribute to the increase in label accuracy because of the karaka labels in Paninian annotation scheme (Begum et al., 2008).

Table 9 also lists the performance of MALT parser with gold morph features (as annotated in HTB). It may be noted that, predicted morph features had similar effect on hindi dependency parsing as of gold features which is desirable making SMA usable for real scenario applications.

Analysis	N	V	PSP	JJ	PRP
Lemma	98.50	94.28	89.41	97.99	98.78
Gender	93.30	95.34	98.93	98.42	94.24
Number	96.26	89.67	96.45	96.26	88.98
Person	98.58	85.28	99.45	99.57	90.94
Case	94.67	98.95	93.26	83.76	95.90
N:Noun, V:Verb, PSP:postposition, JJ:adjective, PRP:pronoun					

Table 13: Overall accuracies for words (by POS tags)

Analysis	Baseline	SMA
Gender	57.09	73.09
Number	76.79	85.71
Person	65.76	77.93
Case	33.62	89.05

Table 14: Evaluation of SMA in a challenging scenario: training data consists only of words analyzed by PBA and test data consists of remaining unanalyzed words.

Analysis	Overall	OOV
Gender	95.68	80.41
Number	94.97	90.30
Person	96.09	96.17
Case	94.61	88.19

Table 15: Accuracy of SMA with auto POS tags

8 Extending the work to Telugu and Urdu

We shall look at how SMA performs in prediciting GNP and case for other morphologically rich Indian languages: Telugu and Urdu. At this stage, we have not done any language-dependent engineering effort

Language	#Sentences	#Words
Urdu	5230	68588
Telugu	1600	6321

Table 16: Telugu and Urdu Treebank Statistics

Analysis	Telugu		Urdu	
	Overall	OOV	Overall	OOV
Gender	96.49	89.85	89.14	88.18
Number	90.65	75.13	91.62	91.35
Person	94.82	85.79	93.37	95.53
Case	96.49	89.34	85.49	79.01

Table 17: SMA for other Mor-FOW languages: Telugu and Urdu

in improving the results rather we want to see how well the system works for other languages using the minimalistic feature set employed for Hindi morphological analysis.

Telugu Treebank was released for ICON 2010 Shared Task(Husain et al., 2010) and a modified version of that data is used for our experiments. Urdu Treebank which is still under development at IIIT Hyderabad[10] is used for experiments related to Urdu morph analysis. Refer table 16 for treebank statistics.

Table 17 shows the evaluation results for Telugu and Urdu.

9 Conclusion and Future work

In conclusion, SMA is a robust state-of-the-art statistical morphological analyzer which outperforms previous analyzers for Hindi by a considerable margin. SMA achieved an accuracy of 63.06% for lemma, gender, number, person and case whereas PBA and Morfette are 34.89% and 51.52% accurate respectively. With the predicted morphological attributes by SMA, we achieve a labeled attachment score of 89.41 while without these morphological attributes the parsing accuracy drops to 87.75.

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.

Performance of Morfette is comparable to SMA and for lemma prediction in the case of OOV words, Morfette outperforms SMA. We plan to build a hybrid system whose feature set includes features from both the systems.

[10] iiit.ac.in

References

Bharat Ram Ambati, Samar Husain, Sambhav Jain, Dipti Misra Sharma, and Rajeev Sangal. 2010a. Two methods to incorporate local morphosyntactic features in hindi dependency parsing. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 22–30. Association for Computational Linguistics.

Bharat Ram Ambati, Samar Husain, Joakim Nivre, and Rajeev Sangal. 2010b. 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.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27.

Grzegorz Chrupała, Georgiana Dinu, and Josef Van Genabith. 2008. Learning morphology with morfette.

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.

Samar Husain, Prashanth Mannem, Bharat Ram Ambati, and Phani Gadde. 2010. The icon-2010 tools contest on indian language dependency parsing. *Proceedings of ICON-2010 Tools Contest on Indian Language Dependency Parsing, ICON*, 10:1–8.

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.

Deepak Kumar Malladi and Prashanth Mannem. 2013. Statistical morphological analyzer for hindi. In *Proceedings of 6th International Joint Conference on Natural Language Processing*.

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 VanGenabith, 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.