Large-Coverage Root Lexicon Extraction for Hindi

Cohan Sujay Carlos   **Monojit Choudhury**   **Sandipan Dandapat**
Microsoft Research India
monojitc@microsoft.com

Abstract

This paper describes a method using morphological rules and heuristics, for the automatic extraction of large-coverage lexicons of stems and root word-forms from a raw text corpus. We cast the problem of high-coverage lexicon extraction as one of stemming followed by root word-form selection. We examine the use of POS tagging to improve precision and recall of stemming and thereby the coverage of the lexicon. We present accuracy, precision and recall scores for the system on a Hindi corpus.

1 Introduction

Large-coverage morphological lexicons are an essential component of morphological analysers. Morphological analysers find application in language processing systems for tasks like tagging, parsing and machine translation. While raw text is an abundant and easily accessible linguistic resource, high-coverage morphological lexicons are scarce or unavailable in Hindi as in many other languages (Clément et al., 2004). Thus, the development of better algorithms for the extraction of morphological lexicons from raw text corpora is a task of considerable importance.

A root word-form lexicon is an intermediate stage in the creation of a morphological lexicon. In this paper, we consider the problem of extracting a large-coverage root word-form lexicon for the Hindi language, a highly inflectional and moderately agglutinative Indo-European language spoken widely in South Asia.

Since a POS tagger, another basic tool, was available along with POS tagged data to train it, and since the error patterns indicated that POS tagging could greatly improve the accuracy of the lexicon, we used the POS tagger in our experiments on lexicon extraction.

Previous work in morphological lexicon extraction from a raw corpus often does not achieve very high precision and recall (de Lima, 1998; Oliver and Tadić, 2004). In some previous work the process of lexicon extraction involves incremental or post-construction manual validation of the entire lexicon (Clément et al., 2004; Sagot, 2005; Forsberg et al., 2006; Sagot et al., 2006; Sagot, 2007).

Our method attempts to improve on and extend the previous work by increasing the precision and recall of the system to such a point that manual validation might even be rendered unnecessary. Yet another difference, to our knowledge, is that in our method we cast the problem of lexicon extraction as two subproblems: that of stemming and following it, that of root word-form selection.

The input resources for our system are as follows: a) raw text corpus, b) morphological rules, c) POS tagger and d) word-segmentation labelled data. We output a stem lexicon and a root word-form lexicon.

We take as input a raw text corpus and a set of morphological rules. We first run a stemming algorithm that uses the morphological rules and some heuristics to obtain a stem dictionary. We then create a root dictionary from the stem dictionary.

The last two input resources are optional but when a POS tagger is utilized, the F-score (harmonic mean of precision and recall) of the root lexicon can be as high as 94.6%.

In the rest of the paper, we provide a brief overview of the morphological features of the Hindi language, followed by a description of our method including the specification of rules, the corpora and the heuristics for stemming and root word-form selection. We then evaluate the system with and without the POS tagger.

Proceedings of the 12th Conference of the European Chapter of the ACL, pages 121–129,
Athens, Greece, 30 March – 3 April 2009. ©2009 Association for Computational Linguistics

2 Hindi Orthography and Morphology

There are some features peculiar to Hindi orthography and to the character encoding system that we use. These need to be compensated for in the system. It was also found that Hindi's inflectional morphology has certain characteristics that simplify the word segmentation rules.

2.1 Orthography

Hindi is written in the partially-phonemic Devanagari script. Most consonant clusters that occur in the language are represented by characters and ligatures, while a very few are represented as diacritics. Vowels that follow consonants or consonant clusters are marked with diacritics. However, each consonant in the Devanagari script also carries an implicit vowel a[1] unless its absence is marked by a special diacritic "halant". Vowels are represented by vowel characters when they occur at the head of a word or after another vowel.

The y sound sometimes does not surface in the pronunciation when it occurs between two vowels. So suffixes where the y is followed by e or I can be written in two ways, with or without the y sound in them. For instance the suffix ie can also be written as iye.

Certain stemming rules will therefore need to be duplicated in order to accommodate the different spelling possibilities and the different vowel representations in Hindi. The character encoding also plays a small but significant role in the ease of stemming of Hindi word-forms.

2.2 Unicode Representation

We used Unicode to encode Hindi characters. The Unicode representation of Devanagari treats simple consonants and vowels as separate units and so makes it easier to match substrings at consonant-vowel boundaries. Ligatures and diacritical forms of consonants are therefore represented by the same character code and they can be equated very simply.

However, when using Unicode as the character encoding, it must be borne in mind that there are different character codes for the vowel diacritics and for the vowel characters for one and the same vowel sound, and that the long and short forms of the vowels are represented by different codes. These artifacts of the character encoding need to be compensated for when using substring matches to identify the short vowel sound as being part of the corresponding prolonged vowel sound and when stemming.

Word Form	Derivational Segmentation	Root
karnA	kar + nA	kar
karAnA	kar + A + nA	kar
karvAnA	kar + vA + nA	kar

Word Form	Inflectional Segmentation	Root
karnA	kar + nA	kar
karAnA	karA + nA	karA
karvAnA	karvA + nA	karvA

Table 1: Morpheme Segmentation

laDkA	Nominative	Oblique
Singular	laDkA	laDke
Plural	laDke	laDkon

laDkI	Nominative	Oblique
Singular	laDkI	laDkI
Plural	laDkI	laDkiyAn

Table 2: Sample Paradigms

2.3 Morphology

The inflectional morphology of Hindi does not permit agglutination. This helps keep the number of inflectional morphological rules manageable. However, the derivational suffixes are agglutinative, leading to an explosion in the number of root word-forms in the inflectional root lexicon.

The example in Table 1 shows that verbs can take one of the two causative suffixes A and vA. These being derivational suffixes are not stemmed in our system and cause the verb lexicon to be larger than it would have otherwise.

2.4 Paradigms

Nouns, verbs and adjectives are the main POS categories that undergo inflection in Hindi according to regular paradigm rules.

For example, Hindi nouns inflect for case and number. The inflections for the paradigms that the words laDkA (meaning boy) and laDkI (meaning girl) belong to are shown in Table 2. The root word-forms are laDkA and laDkI respectively (the singular and nominative forms).

[1] In the discussion in Section 2 and in Table 1 and Table 2, we have used a loose phonetic transcription that resembles ITRANS (developed by Avinash Chopde http://www.aczoom.com/itrans/).

Hindi verbs are inflected by gender, number, person, mood and tense. Hindi adjectives take inflections for gender and case. The number of inflected forms in different POS categories varies considerably, with verbs tending to have a lot more inflections than other POS categories.

3 System Description

In order to construct a morphological lexicon, we used a rule-based approach combined with heuristics for stem and root selection. When used in concert with a POS tagger, they could extract a very accurate morphological lexicon from a raw text corpus. Our system therefore consists of the following components:

1. A raw text corpus in the Hindi language large enough to contain a few hundred thousand unique word-forms and a smaller labelled corpus to train a POS tagger with.
2. A list of rules comprising suffix strings and constraints on the word-forms and POS categories that they can be applied to.
3. A stemmer that uses the above rules, and some heuristics to identify and reduce inflected word-forms to stems.
4. A POS tagger to identify the POS category or categories that the word forms in the raw text corpus can belong to.
5. A root selector that identifies a root word-form and its paradigm from a stem and a set of inflections of the stem.

The components of the system are described in more detail below.

3.1 Text Corpora

Rules alone are not always sufficient to identify the best stem or root for a word-form, when the words being stemmed have very few inflectional forms or when a word might be stemmed in one of many ways. In that case, a raw text corpus can provide important clues for identifying them.

The raw text corpus that we use is the Web-Duniya corpus which consists of 1.4 million sentences of newswire and 21.8 million words. The corpus, being newswire, is clearly not balanced. It has a preponderance of third-person forms whereas first and second person inflectional forms are under-represented.

Name	POS	Paradigm Suffixes	Root
laDkA	noun	{‘A’,‘e’,‘on’}	‘A’
laDkI	noun	{‘I’,‘iyAn’}	‘I’
dho	verb	{‘’,‘yogI’,‘nA’,...}	‘’
chal	verb	{‘’,‘ogI’,‘nA’,...}	‘’

Table 3: Sample Paradigm Suffix Sets

Since Hindi word boundaries are clearly marked with punctuation and spaces, tokenization was an easy task. The raw text corpus yielded approximately 331000 unique word-forms. When words beginning with numbers were removed, we were left with about 316000 unique word-forms of which almost half occurred only once in the corpus.

In addition, we needed a corpus of 45,000 words labelled with POS categories using the IL-POST tagset (Sankaran et al., 2008) for the POS tagger.

3.2 Rules

The morphological rules input into the system are used to recognize word-forms that together belong to a paradigm. Paradigms can be treated as a set of suffixes that can be used to generate inflectional word-forms from a stem. The set of suffixes that constitutes a paradigm defines an equivalence class on the set of unique word-forms in the corpus.

For example, the laDkA paradigm in Table 2 would be represented by the set of suffix strings {‘A’, ‘e’, ‘on’} derived from the word-forms laDkA, laDke and laDkon. A few paradigms are listed in Table 3.

The suffix set formalism of a paradigm closely resembles the one used in a previous attempt at unsupervised paradigm extraction (Zeman, 2007) but differs from it in that Zeman (2007) considers the set of word-forms that match the paradigm to be a part of the paradigm definition.

In our system, we represent the morphological rules by a list of suffix add-delete rules. Each rule in our method is a five-tuple $\{\alpha, \beta, \gamma, \delta, \epsilon\}$ where:

- α is the suffix string to be matched for the rule to apply.
- β is the portion of the suffix string after which the stem ends.
- γ is a POS category in which the string α is a valid suffix.

α	β	γ	δ	ϵ
‘A’	‘’	Noun	N1	‘A’
‘on’	‘’	Noun	N1,N3	‘A’
‘e’	‘’	Noun	N1	‘A’
‘oyogI’	‘o’	Verb	V5	‘o’

Table 4: Sample Paradigm Rules

Word Form	α Match	Stem	Root
laDkA	laDk + A	laDk	laDkA
laDkon	laDk + on	laDk	laDkA
laDke	laDk + e	laDk	laDkA
dhoyogI	dh + oyogI	dh + o	dho

Table 5: Rule Application

- δ is a list of paradigms that contain the suffix string α.
- ϵ is the root suffix

The sample paradigm rules shown in Table 4 would match the words laDkA, laDkon, laDke and dhoyogI respectively and cause them to be stemmed and assigned roots as shown in Table 5.

The rules by themselves can identify word-and-paradigm entries from the raw text corpus if a sufficient number of inflectional forms were present. For instance, if the words laDkA and laDkon were present in the corpus, by taking the intersection of the paradigms associated with the matching rules in Table 4, it would be possible to infer that the root word-form was laDkA and that the paradigm was N1.

We needed to create about 300 rules for Hindi. The rules could be stored in a list indexed by the suffix in the case of Hindi because the number of possible suffixes was small. For highly agglutinative languages, such as Tamil and Malayalam, which can have thousands of suffixes, it would be necessary to use a Finite State Machine representation of the rules.

3.3 Suffix Evidence

We define the term ‘suffix evidence’ for a potential stem as the number of word-forms in the corpus that are composed of a concatenation of the stem and any valid suffix. For instance, the suffix evidence for the stem laDk is 2 if the word-forms laDkA and laDkon are the only word-forms with the prefix laDk that exist in the corpus and A and on are both valid suffixes.

BSE	Word-forms	Accuracy
1	20.5%	79%
2	20.0%	70%
3	13.2%	70%
4	10.8%	81%
5 & more	35.5%	80%

Table 6: % Frequency and Accuracy by BSE

BSE	Nouns	Verbs	Others
1	292	6	94
2	245	2	136
3	172	15	66
4	120	16	71
5 & more	103	326	112

Table 7: Frequency by POS Category

Table 6 presents word-form counts for different suffix evidence values for the WebDuniya corpus. Since the real stems for the word-forms were not known, the prefix substring with the highest suffix evidence was used as the stem. We shall call this heuristically selected stem the best-suffix-evidence stem and its suffix evidence as the best-suffix-evidence (BSE).

It will be seen from Table 6 that about 20% of the words have a BSE of only 1. Altogether about 40% of the words have a BSE of 1 or 2. Note that all words have a BSE of atleast 1 since the empty string is also considered a valid suffix. The fraction is even higher for nouns as shown in Table 7.

It must be noted that the number of nouns with a BSE of 5 or more is in the hundreds only because of erroneous concatenations of suffixes with stems. Nouns in Hindi do not usually have more than four inflectional forms.

The scarcity of suffix evidence for most word-forms poses a huge obstacle to the extraction of a high-coverage lexicon because :

1. There are usually multiple ways to pick a stem from word-forms with a BSE of 1 or 2.
2. Spurious stems cannot be detected easily when there is no overwhelming suffix evidence in favour of the correct stem.

3.4 Gold Standard

The gold standard consists of one thousand word-forms picked at random from the intersection of the unique word-forms in the unlabelled WebDuniya corpus and the POS labelled corpus. Each word-form in the gold standard was manually examined and a stem and a root word-form found for it.

For word-forms associated with multiple POS categories, the stem and root of a word-form were listed once for each POS category because the segmentation of a word could depend on its POS category. There were 1913 word and POS category combinations in the gold standard.

The creation of the stem gold standard needed some arbitrary choices which had to be reflected in the rules as well. These concerned some words which could be stemmed in multiple ways. For instance, the noun `laDkI` meaning 'girl' could be segmented into the morphemes `laDk` and `I` or allowed to remain unsegmented as `laDkI`. This is because by doing the former, the stems of both `laDkA` and `laDkI` could be conflated whereas by doing the latter, they could be kept separate from each other. We arbitrarily made the choice to keep nouns ending in `I` unsegmented and made our rules reflect that choice.

A second gold standard consisting of 1000 word-forms was also created to be used in evaluation and as training data for supervised algorithms. The second gold standard contained 1906 word and POS category combinations. Only word-forms that did not appear in the first gold standard were included in the second one.

3.5 Stemmer

Since the list of valid suffixes is given, the stemmer does not need to discover the stems in the language but only learn to apply the right one in the right place. We experimented with three heuristics for finding the right stem for a word-form. The heuristics were:

- Longest Suffix Match (LSM) - Picking the longest suffix that can be applied to the word-form.

- Highest Suffix Evidence (HSE) - Picking the suffix which yields the stem with the highest value for suffix evidence.

- Highest Suffix Evidence with Supervised Rule Selection (HSE + Sup) - Using labelled data to modulate suffix matching.

3.5.1 Longest Suffix Match (LSM)

In the LSM heuristic, when multiple suffixes can be applied to a word-form to stem it, we choose the longest one. Since Hindi has concatenative morphology with only postfix inflection, we only need to find one matching suffix to stem it. It is claimed in the literature that the method of using the longest suffix match works better than random suffix selection (Sarkar and Bandyopadhyay, 2008). This heuristic was used as the baseline for our experiments.

3.5.2 Highest Suffix Evidence (HSE)

In the HSE heuristic, which has been applied before to unsupervised morphological segmentation (Goldsmith, 2001), stemming (Pandey and Siddiqui, 2008), and automatic paradigm extraction (Zeman, 2007), when multiple suffixes can be applied to stem a word-form, the suffix that is picked is the one that results in the stem with the highest suffix evidence. In our case, when computing the suffix evidence, the following additional constraint is applied: all the suffixes used to compute the suffix evidence score for any stem must be associated with the same POS category.

For example, the suffix `yon` is only applicable to nouns, whereas the suffix `ta` is only applicable to verbs. These two suffixes will therefore never be counted together in computing the suffix evidence for a stem. The algorithm for determining the suffix evidence computes the suffix evidence once for each POS category and then returns the maximum.

In the absence of this constraint, the accuracy drops as the size of the raw word corpus increases.

3.5.3 HSE and Supervised Rule Selection (HSE + Sup)

The problem with the aforementioned heuristics is that there are no weights assigned to rules. Since the rules for the system were written to be as general and flexible as possible, false positives were commonly encountered. We propose a very simple supervised learning method to circumvent this problem.

The training data used was a set of 1000 word-forms sampled, like the gold standard, from the unique word-forms in the intersection of the raw text corpus and the POS labelled corpus. The set of word-forms in the training data was disjoint from the set of word-forms in the gold standard.

Rules	Accur	Prec	Recall	F-Score
Rules1	73.65%	68.25%	69.4%	68.8%
Rules2	75.0%	69.0%	77.6%	73.0%

Table 8: Comparison of Rules

Gold 1	Accur	Prec	Recall	F-Score
LSM	71.6%	65.8%	66.1%	65.9%
HSE	76.7%	70.6%	77.9%	74.1%
HSE+Sup	78.0%	72.3%	79.8%	75.9%

Gold 2	Accur	Prec	Recall	F-Score
LSM	75.7%	70.7%	72.7%	71.7%
HSE	75.0%	69.0%	77.6%	73.0%
HSE+Sup	75.3%	69.3%	78.0%	73.4%

Table 9: Comparison of Heuristics

The feature set consisted of two features: the last character (or diacritic) of the word-form, and the suffix. The POS category was an optional feature and used when available. If the number of incorrect splits exceeded the number of correct splits given a feature set, the rule was assigned a weight of 0, else it was given a weight of 1.

3.5.4 Comparison

We compare the performance of our rules with the performance of the Lightweight Stemmer for Hindi (Ramanathan and Rao, 2003) with a reported accuracy of 81.5%. The scores we report in Table 8 are the average of the LSM scores on the two gold standards. The stemmer using the standard rule-set (Rules1) does not perform as well as the Lightweight Stemmer. We then handcrafted a different set of rules (Rules2) with adjustments to maximize its performance. The accuracy was better than Rules1 but not quite equal to the Lightweight Stemmer. However, since our gold standard is different from that used to evaluate the Lightweight Stemmer, the comparison is not necessarily very meaningful.

As shown in Table 9, in F-score comparisons, HSE seems to outperform LSM and HSE+Sup seems to outperform HSE, but the improvement in performance is not very large in the case of the second gold standard. In terms of accuracy scores, LSM outperforms HSE and HSE+Sup when evaluated against the second gold standard.

POS	Correct	Incorrect	POS Errors
Noun	749	231	154
Verb	324	108	0
Adjective	227	49	13
Others	136	82	35

Table 10: Errors by POS Category

3.5.5 Error Analysis

Table 10 lists the number of correct stems, incorrect stems, and finally a count of those incorrect stems that the HSE+Sup heuristic would have gotten right if the POS category had been available. From the numbers it appears that a sizeable fraction of the errors, especially with noun word-forms, is caused when a suffix of the wrong POS category is applied to a word-form. Moreover, prior work in Bangla (Sarkar and Bandyopadhyay, 2008) indicates that POS category information could improve the accuracy of stemming.

Assigning POS categories to word-forms requires a POS tagger and a substantial amount of POS labelled data as described below.

3.5.6 POS Tagging

The POS tagset used was the hierarchical tagset IL-POST (Sankaran et al., 2008). The hierarchical tagset supports broad POS categories like nouns and verbs, less broad POS types like common and proper nouns and finally, at its finest granularity, attributes like gender, number, case and mood.

We found that with a training corpus of about 45,000 tagged words (2366 sentences), it was possible to produce a reasonably accurate POS tagger[2], use it to label the raw text corpus with broad POS tags, and consequently improve the accuracy of stemming. For our experiments, we used both the full training corpus of 45,000 words and a subset of the same consisting of about 20,000 words. The POS tagging accuracies obtained were approximately 87% and 65% respectively.

The reason for repeating the experiment using the 20,000 word subset of the training data was to demonstrate that a mere 20,000 words of labelled data, which does not take a very great amount of

[2] The Part-of-Speech tagger used was an implementation of a Cyclic Dependency Network Part-of-Speech tagger (Toutanova et al., 2003). The following feature set was used in the tagger: tag of previous word, tag of next word, word prefixes and suffixes of length exactly four, bigrams and the presence of numbers or symbols.

time and effort to create, can produce significant improvements in stemming performance.

In order to assign tags to the words of the gold standard, sentences from the raw text corpus containing word-forms present in the gold standard were tagged using a POS tagger. The POS categories assigned to each word-form were then read off and stored in a table.

Once POS tags were associated with all the words, a more restrictive criterion for matching a rule to a word-form could be used to calculate the BSE in order to determine the stem of the word-form. When searching for rules, and consequently the suffixes, to be applied to a word-form, only rules whose γ value matches the word-form's POS category were considered. We shall call the HSE heuristic that uses POS information in this way HSE+Pos.

3.6 Root Selection

The stem lexicon obtained by the process described above had to be converted into a root word-form lexicon. A root word-form lexicon is in some cases more useful than a stem lexicon, for the following reasons:

1. Morphological lexicons are traditionally indexed by root word-forms

2. Multiple root word-forms may map to one stem and be conflated.

3. Tools that use the morphological lexicon may expect the lexicon to consist of roots instead of stems.

4. Multiple root word-forms may map to one stem and be conflated.

5. Stems are entirely dependent on the way stemming rules are crafted. Roots are independent of the stemming rules.

The stem lexicon can be converted into a root lexicon using the raw text corpus and the morphological rules that were used for stemming, as follows:

1. For any word-form and its stem, list all rules that match.

2. Generate all the root word-forms possible from the matching rules and stems.

3. From the choices, select the root word-form with the highest frequency in the corpus.

Relative frequencies of word-forms have been used in previous work to detect incorrect affix attachments in Bengali and English (Dasgupta and Ng, 2007). Our evaluation of the system showed that relative frequencies could be very effective predictors of root word-forms when applied within the framework of a rule-based system.

4 Evaluation

The goal of our experiment was to build a high-coverage morphological lexicon for Hindi and to evaluate the same. Having developed a multi-stage system for lexicon extraction with a POS tagging step following by stemming and root word-form discovery, we proceeded to evaluate it as follows.

The stemming and the root discovery module were evaluated against the gold standard of 1000 word-forms. In the first experiment, the precision and recall of stemming using the HSE+Pos algorithm were measured at different POS tagging accuracies.

In the second experiment the root word-form discovery module was provided the entire raw word corpus to use in determining the best possible candidate for a root and tested using the gold standard. The scores obtained reflect the performance of the overall system.

For stemming, the recall was calculated as the fraction of stems and suffixes in the gold standard that were returned by the stemmer for each word-form examined. The precision was calculated as the fraction of stems and suffixes returned by the stemmer that matched the gold standard. The F-score was calculated as the harmonic mean of the precision and recall.

The recall of the root lexicon was measured as the fraction of gold standard roots that were in the lexicon. The precision was calculated as the fraction of roots in the lexicon that were also in the gold standard. Accuracy was the percentage of gold word-forms' roots that were matched exactly.

In order to approximately estimate the accuracy of a stemmer or morphological analyzer that used such a lexicon, we also calculated the accuracy weighted by the frequency of the word-forms in a small corpus of running text. The gold standard tokens were seen in this corpus about 4400 times. We only considered content words (nouns, verbs, adjectives and adverbs) in this calculation.

Gold1	**Accur**	**Prec**	**Recall**	**F-Sco**
POS	86.7%	82.4%	86.2%	84.2%
Sup+POS	88.2%	85.2%	87.3%	86.3%
Gold2	**Accur**	**Prec**	**Recall**	**F-Sco**
POS	81.8%	77.8%	82.0%	79.8%
Sup+POS	83.5%	80.2%	82.6%	81.3%

Table 11: Stemming Performance Comparisons

Gold 1	**Accur**	**Prec**	**Recall**	**F-Sco**
No POS	76.7%	70.6%	77.9%	74.1%
65% POS	82.3%	77.5%	81.4%	79.4%
87% POS	85.4%	80.8%	85.1%	82.9%
Gold POS	86.7%	82.4%	86.2%	84.2%

Table 12: Stemming Performance at Different POS Tagger Accuracies

5 Results

The performance of our system using POS tag information is comparable to that obtained by Sarkar and Bandyopadhyay (2008). Sarkar and Bandyopadhyay (2008) obtained stemming accuracies of 90.2% for Bangla using gold POS tags. So in the comparisons in Table 11, we use gold POS tags (row two) and also supervised learning (row three) using the other gold corpus as the labelled training corpus. We present the scores for the two gold standards separately. It must be noted that Sarkar and Bandyopadhyay (2008) conducted their experiments on Bangla, and so the results are not exactly comparable.

We also evaluate the performance of stemming using HSE with POS tagging by a real tagger at two different tagging accuracies - approximately 65% and 87% - as shown in Table 12. We compare the performance with gold POS tags and a baseline system which does not use POS tags. We do not use labelled training data for this section of the experiments and only evaluate against the first gold standard.

Table 13 compares the F-scores for root discovery at different POS tagging accuracies against a baseline which excludes the use of POS tags altogether. There seems to be very little prior work that we can use for comparison here. To our knowledge, the closest comparable work is a system built by Oliver and Tadić (2004) in order to enlarge a Croatian Morphological Lexicon. The overall performance reported by Tadić et al was as follows: (precision=86.13%, recall=35.36%, F1=50.14%).

Gold 1	**Accur**	**Prec**	**Recall**	**F-Sco**
No POS	71.7%	77.6%	78.8%	78.1%
65% POS	82.5%	87.2%	88.9%	88.0%
87% POS	87.0%	94.1%	95.3%	94.6%
Gold POS	89.1%	95.4%	97.9%	96.6%

Table 13: Root Finding Accuracy

Gold 1	Stemming	Root Finding
65% POS	85.6%	87.0%
87% POS	87.5%	90.6%
Gold POS	88.5%	90.2%

Table 14: Weighted Stemming and Root Finding Accuracies (only Content Words)

Lastly, Table 14 shows the accuracy of stemming and root finding weighted by the frequencies of the words in a running text corpus. This was calculated only for content words.

6 Conclusion

We have described a system for automatically constructing a root word-form lexicon from a raw text corpus. The system is rule-based and utilizes a POS tagger. Though preliminary, our results demonstrate that it is possible, using this method, to extract a high-precision and high-recall root word-form lexicon. Specifically, we show that with a POS tagger capable of labelling word-forms with POS categories at an accuracy of about 88%, we can extract root word-forms with an accuracy of about 87% and a precision and recall of 94.1% and 95.3% respectively.

Though the system has been evaluated on Hindi, the techniques described herein can probably be applied to other inflectional languages. The rules selected by the system and applied to the word-forms also contain information that can be used to determine the paradigm membership of each root word-form. Further work could evaluate the accuracy with which we can accomplish this task.

7 Acknowledgements

We would like to thank our colleagues Priyanka Biswas, Kalika Bali and Shalini Hada, of Microsoft Research India, for their assistance in the creation of the Hindi root and stem gold standards.

References

Lionel Clément, Benoît Sagot and Bernard Lang. 2004. Morphology based automatic acquisition of large-coverage lexica. In *Proceedings of LREC 2004*, Lisbon, Portugal.

Sajib Dasgupta and Vincent Ng. 2007. High-Performance, Language-Independent Morphological Segmentation. In *Main Proceedings of NAACL HLT 2007*, Rochester, NY, USA.

Markus Forsberg, Harald Hammarström and Aarne Ranta. 2006. Morphological Lexicon Extraction from Raw Text Data. In *Proceedings of the 5th International Conference on Advances in Natural Language Processing, FinTAL*, Finland.

John A. Goldsmith. 2001. Linguistica: An Automatic Morphological Analyzer. In *Arika Okrent and John Boyle, editors, CLS 36: The Main Session, volume 36-1*, Chicago Linguistic Society, Chicago.

Erika de Lima. 1998. Induction of a Stem Lexicon for Two-Level Morphological Analysis. In *Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning, NeMLaP3/CoNLL98, pp 267-268*, Sydney, Australia.

Antoni Oliver, Marko Tadić. 2004. Enlarging the Croatian Morphological Lexicon by Automatic Lexical Acquisition from Raw Corpora. In *Proceedings of LREC 2004*, Lisbon, Portugal.

Amaresh Kumar Pandey and Tanveer J. Siddiqui. 2008. An Unsupervised Hindi Stemmer with Heuristic Improvements. In *Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data, AND 2008, pp 99-105*, Singapore.

A Ramanathan and D. D. Rao. 2003. A Lightweight Stemmer for Hindi. Presented at *EACL 2003*, Budapest, Hungary.

Benoît Sagot. 2005. Automatic Acquisition of a Slovak Lexicon from a Raw Corpus. In *Lecture Notes in Artificial Intelligence 3658, Proceedings of TSD'05*, Karlovy Vary, Czech Republic.

Benoît Sagot. 2007. Building a Morphosyntactic Lexicon and a Pre-Syntactic Processing Chain for Polish. In *Proceedings of LTC 2007*, Poznań, Poland.

Benoît Sagot, Lionel Clément, Éric Villemonte de la Clergerie and Pierre Boullier. 2006. The Lefff 2 Syntactic Lexicon for French: Architecture, Acquisition, Use. In *Proceedings of LREC'06*, Genoa, Italy.

Baskaran Sankaran, Kalika Bali, Monojit Choudhury, Tanmoy Bhattacharya, Pushpak Bhattacharyya, Girish Nath Jha, S. Rajendran, K. Saravanan, L. Sobha and K.V. Subbarao. 2008. A Common Parts-of-Speech Tagset Framework for Indian Languages. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco.

Sandipan Sarkar and Sivaji Bandyopadhyay. 2008. Design of a Rule-based Stemmer for Natural Language Text in Bengali. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, Hyderabad, India.

Kristina Toutanova, Dan Klein, Christopher D. Manning and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependedependency Network In *Proceedings of HLT-NAACL 2003 pages 252-259.*

Daniel Zeman. 2007. Unsupervised Acquisition of Morphological Paradigms from Tokenized Text. In *Working Notes for the Cross Language Evaluation Forum CLEF 2007 Workshop*, Budapest, Hungary.