Computational Morphology and Syntax for a Resource-Poor Inflectional Language

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Navanath Saharia

Enrolment No. CSP-08-003

Registration No. 35 of 2010

Department of Computer Science and Engineering

School of Engineering, Tezpur University

Tezpur, Assam, India - 784028

January, 2014

Dedicated to my

$\sim \star \sim$

F

A

M

I

L

Y

$\sim \star \sim$

Abstract

In this work, we attempt to design a computational model to analyse the morphology and syntax of resource poor-languages. We primarily experiment on Assamese, a morphologically rich, inflectional and resource-poor Indian language. We subdivide our problem into sub-goals.

1. Finding the root/stem.
2. Assigning grammatical category to the words.
3. Analysis the sentence structure.

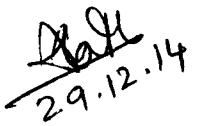
For each sub-goal, we conduct a series of experiments. For stemming, first we design a rule-based method to remove suffixes from given Assamese words. To reduce over-stemming and under-stemming errors, we introduce a dictionary of frequent words. We observe that, for these languages, a dominant proportion of suffixes are single letter and this creates problems during suffix stripping. Finally, we introduce an HMM based hybrid approach to classify the mis-match of the last characters with a set of single letter suffixes. For each word, stemming is performed by computing the most probable path among the four states defined in the HMM. After obtaining encouraging results for Assamese, we use the same approaches to stem text in other Indian languages, viz, Bengali, Bishnupriya Manipuri and Bodo to demonstrate the generality of our method. At each step we measure the stemming accuracy for each language and compare our results with other published works. We have designed an Assamese specific hierarchical tagset and experiment with three PoS tagging methods for Assamese text. The first approach is the rule-based approach to classify noun and verb from raw text. After that, we use a dictionary with the rule-based approach to increase the PoS tagging accuracy. Lastly, we experiment with an HMM based approach to classify words in Assamese text. We obtain 87-90% precision using the HMM based tagger. We also run experiments to group the dependent word into a single

unit. We label each sentence with standard IOB (Inside Outside Beginning) [1] tags and employ Yamcha [2], a supervised support vector based tool to identify and classify multi-word unit from the annotated corpus. We explore three dependency parsing models for Assamese, viz. Link grammar [3] parsing, Malt parsing [4] and MST [5] parsing. We have developed the rule-base and a dictionary for link grammar parser for Assamese. We have also compared the performance of these three parsing models. We have developed a TreeBank-a repository to store the parsed sentences.

Keywords - Stemming, POS tagging, Parsing, Treebank, Assamese, Resource-poor language, Inflectional language

Declaration

I, Navanath Saharia, hereby declare that the thesis entitled **Computational Morphology and Syntax for a Resource-Poor Inflectional Language** submitted to the Department of Computer Science and Engineering under the School of Engineering, Tezpur University, in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy, is based on bona fide work carried out by me. The results embodied in this thesis have not been submitted in part or in full, to any other university or institute for award of any degree or diploma.



(Navanath Saharia)

Heartfelt thanks ...

... to my supervisor Prof. Utpal Sharma from Tezpur University and co-supervisor Prof. Jugal K. Kalita from University of Colorado, Colorado Springs for their constant support, their trust, their valuable feedback, their encouragement and their innumerable advice.

... to the members of the doctoral committee of my research and departmental research committee for their insightful comments and valuable feedback: Prof. Malay A. Dutta and Prof. Dhruba Kumar Bhattacharyya, and all other faculty member of Department of CSE, Tezpur University

... to the members of my thesis review committee and the anonymous reviewer for their comments and feedback.

... to Prof. Jyoti P. Tamuli and Dr. Gitanjali Bez of Gauhati University, Prof. Madhumita Barbora and Dr. Arup K. Nath of Tezpur University for their constant support from the point of linguistics. Dr. Kishori M. Konwar for the fruitful discussions regarding HMM. Prof. Shikhar K. Sarma for his financial and mental support during my work.

... to all the annotators and manual validator involved in parts of this work for their time and expertise, Prof. Smriti K. Sinha, for manual validation of the Bishnupriya Manipuri text; Raju and Guddu for manual validation of the Bodo text.

... to all my friends specifically Kishore, Bijoy, Dushyant, Monower, Padmaja, Himangshu, Nayan, Juwesh da, Praveen da, Hasin, Madhurjya, Gitartha, Arpana, Adity ba, Mancha and all well wishers.

... A very special mention goes to **Maa** and **Deuta**. encouraging and supportive as ever and thanks for trusting me, as always. I certainly do not forget either Sandhya, Aitu or Baba and I want to warmly thank them here for their unconditional and incredible understanding and support.

- Navanath Saharia

Tezpur University
School of Engineering
Department of Computer Science & Engineering
Napaam, Assam, India-784028

Dr. Utpal Sharma
Professor

Email: utpal@tezu.ernet.in
Phone: +91-3712-275107

Certificate

This is to certify that the thesis entitled **"Computational Morphology and Syntax for a Resource-Poor Inflectional Language"** submitted to the School of Engineering, Tezpur University in partial fulfillment for the award of the degree of Doctor of Philosophy in the Department of Computer Science and Engineering is a record of research work carried out by **Mr. Navanath Saharia** under my supervision and guidance.

All help received by him from various sources have been duly acknowledged. No part of this thesis has been submitted elsewhere for award of any other degree.

Signature of Supervisor

29.12.14.

(Dr. Utpal Sharma)
Professor

University of Colorado at Colorado Springs
Department of Computer Science
1420 Austin Bluffs Parkway
Colorado Springs, Colorado 80933-7150

Dr. Jugal K Kalita
Professor

Email: jkalita@uccs.edu
Phone. 719-255-3325

Certificate

This is to certify that the thesis entitled **"Computational Morphology and Syntax for a Resource-Poor Inflectional Language"** submitted to the School of Engineering, Tezpur University in partial fulfillment for the award of the degree of Doctor of Philosophy in the Department of Computer Science and Engineering is a record of research work carried out by **Mr. Navanath Saharia** under my supervision and guidance.

All help received by him from various sources have been duly acknowledged. No part of this thesis has been submitted elsewhere for award of any other degree.

Signature of Co-supervisor

(Dr. Jugal K. Kalita)
Professor

Tezpur University

Certificate

This is to certify that the thesis entitled **"Computational Morphology and Syntax for a Resource-Poor Inflectional Language"** submitted by **Mr. Navanath Saharia** to Tezpur University in the Department of Computer Science and Engineering under the School of Engineering in partial fulfillment for the award of the degree of Doctor of Philosophy in Computer Science and Engineering has been examined by us on ..... and found to be satisfactory.

The committee recommends for award of degree of Doctor of Philosophy.

Signature of Supervisor
Date:

Signature of External Examiner
Date:

Signature of committee member

Contents

Abstract iii

Declaration v

Acknowledgement vi

Table of contents x

List of tables xiv

List of figures xvii

List of abbreviations xix

1 Introduction 1

1.1 Objective 3

1.2 The target language 3

1.3 Contributions of this thesis 4

1.4 Outline of the thesis 5

2 Assamese Corpora 7

2.1 Introduction 8

2.2 Emille corpus 9

2.3 Assamese Pratidin corpus 10
2.4 Wikipedia corpus 10
2.5 Tezu Assamese corpus 11
2.6 Summary 12

3 Stemming in Assamese words **13**

3.1 Introduction 14
3.2 Related work 16
3.3 Language related issues 17
3.4 Approach-1: Rule-based approach 20
3.4.1 Results and discussion 23
3.5 Approach-2: Dictionary look-up-based approach 25
3.5.1 Preparation of dictionary 25
3.5.2 Results and discussion 26
3.6 Approach-3: A Hybrid Approach 28
3.6.1 The HMM model 28
3.6.2 Preparation of training data 30
3.6.3 Results and discussion 32
3.7 Experiments in other languages 33
3.8 Summary 42

4 AsmPoST: Part-of-Speech Tagger for Assamese **44**

4.1 Introduction 45
4.2 Related work 47
4.3 Linguistic issues 49
4.4 Assamese part-of-speech tagset 51

4.4.1 TUTagset-F 52
4.4.2 Xobdo tagset 53
4.4.3 LDC-IL tagset 53
4.4.4 Description of our tagset : *TUTagset-H* 53
4.5 Suffix based noun and verb categorization 59
4.5.1 Noun morphology 60
4.5.2 Verb morphology 60
4.5.3 Morphology driven PoS tagging 60
4.5.4 Results and discussion 61
4.6 Incorporating dictionary to enhance accuracy 62
4.6.1 Results and discussion 64
4.7 HMM based PoS tagging 65
4.7.1 Results and discussion 66
4.8 Experiments in other languages 68
4.9 Tagging Multi-word unit 71
4.9.1 Reduplications 73
4.9.2 Compound nouns 74
4.9.3 Compound and conjunct verbs 76
4.9.4 Results and discussion 76
4.10 Summary 78

5 Parsing Assamese Text **79**

5.1 Introduction 80
5.2 Dependency Grammar Formalism 81
5.3 Related work 83
5.4 Assamese as a relatively free word order language 85

5.5 Parsing of Assamese text 87
5.5.1 Link Grammar parser 87
5.5.1.1 Links and performance analysis 89
5.5.2 Malt Parser 89
5.5.2.1 Results and analysis 91
5.5.3 MST Parser 91
5.5.3.1 Results and analysis 92
5.6 Experiments in other languages 93
5.7 tezuBank: Assamese Dependency TreeBank 94
5.8 Summary 98

6 Conclusion **99**

Appendix A Assamese noun and verb inflections **104**

Appendix B Tagset details **107**

B.1 PENN tagset 107
B.2 BNC tagset 109
B.3 Xobdo tagset 111
B.4 TUTaget–F 112
B.5 LDC-IL tagset 118
B.6 BIS tagset 121
B.7 AnnCora tagset 123

Appendix C Rules of link grammar for Assamese **125**

Bibliography **132**

List of Tables

2.1 Statistics of the used/developed corpora. 9

2.2 First fifteen most frequent words from the corpora. 11

3.1 Pattern of inflections in some Indian language and English for the sentence *Ram killed Ravana* irrespective of the word position. 18

3.2 Categories of suffixes with examples 20

3.3 Analysis of error types of Approach 1. 24

3.4 Result obtained using dictionary with Approach 2. 26

3.5 Root words whose final letters match some suffix. 27

3.6 A random survey on occurrence of single letter suffix, multiple suffix and multiple suffix end-with single letter suffix (MS). 27

3.7 An example sentence modelled using our generative model of the text for morphological inflections. 29

3.8 Suffix information in the Assamese training corpus with 3082 words . 31

3.9 Single letter suffix frequency (F) in the Assamese training corpus. 31

3.10 Result obtained using various approaches. 32

3.11 Comparison of stemming by different approaches for Assamese noun root নাটক (natɒk : *drama*). 34

3.12 Comparison of stemming by different approaches for Assamese verb root কৰ (kɒr : to do). 35

3.13 Single letter suffixes in Bengali, Bishnupriya Manipuri and Bodo with examples of inflected words and root words ending with that letter . . 40

3.14 Suffix information in the training corpora. 41

3.15 Results obtained for Assamese, Bengali, Bishnupriya Manipuri and Bodo using various approaches. 41

3.16 Comparison of our result with other approach 42

4.1 Some reported PoS tagging approaches in Indian languages 49

4.2 Personal definitives are inflected for person and number 49

4.3 Formation of derivational noun and verb in Assamese 50

4.4 Assamese hierarchical tagset 55

4.5 Examples of use of হ (ho : *to be*) verb in Assamese. 56

4.6 Formation of compound words in Assamese 60

4.7 Precision, recall and F-measure of Approach-1 62

4.8 Word list information used in dictionary-based PoS tagging. NOM : Nominal modifier 63

4.9 Label-wise error rate obtained by incorporating dictionary 65

4.10 Results compared with other dictionary based work. 65

4.11 PoS tagging results with flat and hierarchical tagset. 67

4.12 Comparison of our PoS tagging result with other HMM based models. . . 67

4.13 PoS tagging results for Assamese, Bengali, Bishnupriya Manipuri and Bodo using suffix based noun verb identification approach 69

4.14 Label-wise obtained error-rates after incorporating dictionary for Assamese, Bengali, Bishnupriya Manipuri and Bodo language 70

4.15 Results obtained by using HMM with flat and hierarchical tagset for Assamese, Bengali, Bishnupriya Manipuri and Bodo language. 71

4.16 Result of automatic extraction of reduplication in annotated and unannotated text 74

5.1 Word order variation table [6]. 81

5.2 Language-wise survey of implemented parsers. 84

5.3 Two constituent sentences. 85

5.4 Statistics of the input sentences for performance evaluation 90

5.5 Statistics of used corpus for training and testing Malt parser 91
5.6 Average accuracy obtained using Malt parser 91
5.7 Average accuracy obtained using the MST parser 92
5.8 Parsing results using Link grammar, Malt parser and MST parser. 93
A.1 Some inflectional forms for the noun root মানুহ (manuh : *man*) 105
A.2 Some inflectional forms of the root verb আছ (asɒ : *to be*) 106
B.1 Penn tagset 108
B.2 BNC basic tagset 110
B.3 Xobdo tagset 111
B.4 TUTaget−F 117
B.5 LDC-IL tagset 119
B.6 Attribute and their values in LDC-IL Tagset. 120
B.7 BIS tagset 122
B.8 AnnCora tagset 123
B.9 Attribute and their values 124

List of Figures

2.1 Structure of XML corpus in Tezu Assamese corpus. 12

3.1 Impact of accuracy with increasing dictionary size 26

4.1 Example output 64

4.2 Architecture of rule based reduplication identifier for unannotated corpus. 74

5.1 Dependency graph for sentence "I am a student of Tezpur university" . . 81

5.2 Dependency structure for the sentence "মই নতুন কিতাপ কিনিছোঁ।" (mɔi nɔtun kitap kinisũ.) 81

5.3 Phrase structure for the sentence "Quickly go with dad" 82

5.4 Dependency structure for the sentence "Quickly go with dad" 82

5.5 Dependency graph for the simple sentence দেউতা বজাৰলৈ গ'ল (deuta bɒzarɔlɔi gɔl) 88

5.6 Dependency graph for the sentence দেউতা দেওবৰীয়া হাটলৈ চাইকেলেৰে গ'ল (deuta deobɒrija hatɒloi saikelere gɔl) 88

5.7 Dependency graph for the sentence দেউতাই ৰঙা কামিজটো পিন্ধি দেওবৰীয়া হাটলৈ গ'ল (deutai rɒŋa kamizto pind^{h}i deobɒrija hatɒloi gɔl) 88

5.8 Dependency graph for the sentence মই ভাত খাই খেলিবলৈ গ'লোঁ (mɔi b^{h}at k^{h}ai k^{h}ɛlibɔlɔi gɔlo) 88

5.9 Linking requirements of nominal modifiers, adverbs and nouns 89

5.10 Dependency graph for sentence 1. 94

5.11 Dependency graph for sentence 2. 94

C.1 Dependency graph for the sentence ৰঙা আঁচ থকা কামিজটো লেতেৰা (rɔŋa ãs t^{h}ɔka kamizto lɛtɛra) 126

C.2 Dependency graph for the sentence মই আৰু তুমি তাঁলৈ যাম। (mɔi aru tumi talɔi zam.) 127

C.3 Dependency graph for the sentence একৰ পিছত দুই আহে। (ɛkɔr pisɔt dui ahe) 128

List of abbreviations

	A
ACM	Acquisitive Case marker
ANC	American National Corpus
ASCII	American Standard Code for Information Interchange
as/asm	Assamese language
	B
bd	Bodo language
bn	Bengali language
BNC	British National Corpus
bpy	Bishnupriya Manipuri language
	C
CCG	Combinatory Categorial Grammar
CFG	Context Free Grammar
CIIL	Central Institute of Indian languages
CKY	Cocke-Kasami-Younger
CL	Computational Linguistics
CLE	Chu–Liu-Edmonds' algorithm
CM	Case Marker
CRF	Conditional Random Fields
	D
DG	Dependency Grammar
DM	Definitive Marker
	E
EM	Emphatic Marker
EMILLE	Enabling Minority Language Engineering
en	English language
EXT	Extra
	F
F	Frequency
F	F-Score

	G
GCM	Genitive Case marker
	H
HTML	Hyper Text Mark-up Language
HMM	Hidden Markov Model
HPSG	Head-Driven Phrase Structure Grammar
	I
IPA	International Phonetics Alphabets
IOB	Inside Outside Beginning
IR	Information Retrieval
	L
LAS	Labeled Attachment Score
LCM	Locative Case Marker
LDC-IL	Linguistic Data Consortium for Indian Languages
LFG	Lexical Functional Grammar
LLC	London-Lund Corpus
LG	Link Grammar
LS	Labeled Score
	M
MLP	Multilayer Perceptron
MST	Maximum Spanning Tree parser
MWE	Multi Word Expression
MWU	Multi Word Unit
	N
NCM	Nominative Case Marker
NER	Named Entity Recognition
NLP	Natural Language Processing
NN	Noun
NOM	Nominal Modifiers
NP	Noun Phrase
NUM	Number
	O
OBJ	Object
OOV	Out of Vocabulary
OSV	Object Subject Verb
OVS	Object Verb Subject
	P
P	Precision
PG	Paninian Grammar
PL	Plural marker

PoS	Part-of-Speech
PoST	Part-of-Speech Tagging
PN	Pronoun
PP	Prepositional Phrase
PSP	Post-positions
PUN	Punctuation

Q

QW	Question Word

R

R	Recall
RB	Adverb
RDP	Reduplication
RP	Particle

S

SOV	Subject Object Verb
SUB	Subject
SVM	Support Vector Machine
SVO	Subject Verb Object

T

TAG	Tree-Adjoining Grammar
TAM	Tense, Aspect and Modality
TBL	Transformation based learning
TnT	Trigrams 'n' Tags

U

UAS	Unlabeled Attachment Score
UNK	Unknown word
UPH	Unknown Proper Noun handling accuracy
UTF-8	UCS Transformation Format 8 bit
UWH	Unknown Word handling accuracy

V

VB	Verb
VP	Verb Phrase
VOS	Verb Object Subject
VSO	Verb Subject Object

X

XCES	XML Corpus Encoding Standard
XML	Xtended Mark-up Language

Chapter 1

Introduction

"·····Good Morning!" said Bilbo, and he meant it. The sun was shining, and the grass was very green. But Gandalf looked at him from under long bushy eyebrows that stuck out further than the brim of his shady hat.
"What do you mean?" he said. Do you wish me a good morning, or mean that it is a good morning whether I want it or not; or that you feel good this morning; or that it is a morning to be good on?
"All of them at once", said Bilbo.·····

– The Hobbit, J. R. R. Tolkien (1892 - 1973)

Using a language to represent or convey information comes "naturally" to a human, but the use of human language, actually, is far from trivial. The term *computational linguistics* refers to the inter-disciplinary field at the intersection of linguistics, phonetics, computer science, cognitive science, artificial intelligence and formal logic, frequently assisted by statistical techniques [7]. Analysing or processing linguistic phenomena necessitates viewing natural language text in a "structured way". The structured approach to the study of language has given birth to three main areas of linguistics – the study of the arrangement of the basic units of the language following basic rules, the study of the meaning that each word bears, and the study of the context that may change the meaning of a word. Each area or sub-area of language processing (for instance, stemming–finding the root, stem, or base form from an inflected word; identification of grammatical category of a word) needs analysis of linguistic phenomena in a computational environment supported by knowledge of traditional grammar.

A number of approaches to solve specific problems have been proposed and tried. Researchers have tried to address general issues as well as language specific ones. This thesis starts out with a fundamental question–

> *"Are language related techniques developed for one language equally applicable to other languages? Should new techniques be developed that are more appropriate for the individual characteristics of a specific language?"*

The problems we tackle in this work have been amply discussed in the context of a language like English. However, there are many language specific issues that need to be addressed. In addition, we believe that there are easier and more efficient ways to handle issues better than existing techniques for a specific language. From the perspective of the language we are interested in, the problems investigated in this thesis have not been studied. Our experiments attempt to answer the question given above in the context of Assamese. We develop a computational linguistic model of Assamese. In some situations, we extend the proposed techniques to neighboring languages to generalize our approach.

Though Assamese is one of the scheduled national languages of India, little computational work has been done so far for this language. Developing a language-processing tool is not a straightforward process due to the nature of natural languages and the manner in which language is represented at various levels. There are ambiguities at each phase of representation and processing. Since computational linguistics is new and there are substantial variations in terms of rules or dialects, there arises the problem of standardization such as the standardization of morphological classes and the standardization

of a PoS tagset. Overcoming all these barriers requires a lot of effort. This research investigates methodologies to deal with the problems of stemming, PoS tagging, identification of multi-word units and parsing of Assamese text.

1.1 Objective

The primary objective of this thesis is to develop an approach for parsing Assamese text. Early in our research, we realized that the basic software modules and resources required to parse texts are not adequately available for Assamese. Therefore, our work covers the prior tasks essential for parsing, too. We define three interrelated sub-objectives of our work.

- **Module: 1** - Finding the root or stem of a word,
- **Module: 2** - Finding the grammatical category, and
- **Module: 3** - Finding sentence structures

To perform our computational experiments, the basic need is a corpus of Assamese. For our experiments, we use four different collections of texts – (a) Assamese Pratidin corpus[1]; (b) EMILLE corpus[2]; (c) Wikipedia corpus[3]; (d) Tezu Assamese corpus[4]. Except the Assamese Pratidin corpus, others are encoded in UTF-8 format. We have developed a converter to convert ASCII based normalised text of the Assamese Pratidin corpus to UTF-8.

1.2 The target language

In this section, we provide a brief relevant linguistic background of our target language, Assamese. For comparative study and to generalize our proposed approach, we also work with Bengali, Bishnupriya Manipuri and Bodo, in some places. The Assamese language is a member of the Indo-Aryan language family and spoken in the north eastern part of India and its neighbouring regions. The word formation process of Assamese includes

[1]Developed by U. Sharma, Tezpur University. ,
[2]http://lancs.ac.uk/fass/projects/corpus/emille
[3]Collected article from http://as.wikipedia.org
[4]Collection of online news and blog articles

inflection, derivation, coinage, clipping, back formation, translation and transliteration. Being an inflectional and agglutinative language, it is morphologically rich. Though it is a relatively free word order language, the dominant word order is subject-object-verb (SOV). In spoken form, Assamese has a number of dialects and in text, a standard form of the language is generally written using the "Assamese script". When we mention Assamese throughout this thesis we primarily mean standard Assamese as opposed to other dialectal variations.

Like most Indian languages, Assamese has been studied infrequently in the global context. It still lacks even a single balanced corpus or basic language processing modules like stemmers and morphological analysers. Before we started our work, we found a few reported computational efforts in Assamese. Among them [8, 9, 10, 11, 12] are the main. Sharma et al. [8] describe an approach to extract stems from affix evidence. An extended version [10, 12] presents an unsupervised approach to learn morphology using corpus text. A spell checker for ASCII based Assamese text is reported by Das et al. [9]. Considering just this set of reported work as our base, we started our experiments to design the three main language processing tools, namely a stemmer, a PoS tagger and a parser that are discussed at length in subsequent chapters.

For transliteration of examples in the thesis, we use an in-house transliteration scheme and also provide representation using the International Phonetics Alphabets (IPA). The general syntax used in our examples is - **word_in_non-roman_script (IPA : *meaning*)**.

1.3 Contributions of this thesis

The most important contributions of our work are given below.

- **Efficient stemming:** We propose a hybrid stemming approach for Assamese and apply our method to some neighbouring languages, viz., Bengali, Bishnupriya Manipuri, and Bodo.
- **PoS tagger:** We design an Assamese-specific hierarchical tagset and experiment with three PoS tagging methods for Assamese. The first approach is the rule-based approach to classify noun and verb from raw text. After that, we use a dictionary with the rule-based approach to increase the PoS tagging accuracy. Lastly, we

experiment with an HMM based approach to classify Assamese text. We obtain 87-90% precision using the HMM based tagger.

- **Extraction of MWU:** We label each sentence with standard IOB (Inside Outside Beginning) [1] tags and employ Yamcha [2], a supervised Support Vector Machine based tool to identify and classify MWU from the annotated corpus.

- **Dependency parsing model:** We explore three dependency parsing models for Assamese, viz. Link grammar [3] parsing, Malt parsing [4] and MST [5] parsing. We develop the rule-base and a dictionary for link grammar parser. We also compare the performance of these three parsing models. We have developed a repository to store the parsed sentences.

1.4 Outline of the thesis

This thesis is organised around four main parts.

Chapter 2 describes the preparation and development of the Assamese corpora used throughout the experiments. For our experiments, we use four different collections of raw text, viz., the EMILLE corpus, the Assamese Pratidin corpus, the Wikipedia corpus and the Tezu Assamese corpus.

Chapter 3 presents a detailed description of stemming in the context of Assamese text. We analyse previous work on stemming and, we extend our experiments to three other Indian languages to generalise our proposed approach. We examine stemming accuracies with three different approaches and analyse the outcomes. First, we design a rule-based approach to remove suffixes from words. To reduce over-stemming and under-stemming errors, we introduce a dictionary of frequent words. We observe that for these languages a dominant portion of suffixes are 1-letter long and these suffixes cause ambiguity during suffix stripping. Finally, we introduce an HMM based hybrid approach to classify the mis-matching of the last character with the single letter suffix set. For each word, stemming is performed by computing the most probable path in the four defined HMM states.

Chapter 4 describes our work on development of Assamese PoS tagger. We study the state-of-the-art in PoS tagging of Assamese. We develop a hierarchical tagset specifically for Assamese, and introduce three approaches for automatic tagging. The first approach describes the identification of noun and verb, since these two are open

class categories like in other Indian languages. The second approach is a dictionary based model that uses a word list and rule-base to determine the grammatical category and the third approach is a Hidden Markov Model based approach. We also describe our approach to identification and processing of multi-word units. We present our experimental results in identification of reduplicatives, compound nouns and compound verbs.

In **Chapter 5**, we describe dependency parsing and its state-of-the-art for Indian languages. We develop a Link grammar and parser for Assamese. We also explore the Malt parser and MST parser for Assamese and discuss evaluation methods. Finally, we discuss the architecture of *Tezu-TreeBank*, the TreeBank for Assamese.

In **Chapter 6**, we conclude this thesis by summarising our major achievements and discussing possible future work.

Chapter 2

Assamese Corpora

"Don't you believe in anything?
Yes, I said. I believe in evidence. I believe in observation, measurement, and reasoning, confirmed by independent observers. I will believe anything, no matter how wild and ridiculous, if there is evidence for it. The wilder and more ridiculous something is, however, the firmer and more solid the evidence will have to be. "
– The Roving Mind, Isaac Asimov (1920 - 1992)

Outline: This chapter presents the following:

1. A brief introduction to the use of corpora in computational linguistic.
2. A brief description of preparing the corpora for experiments.
3. The most frequent words in the corpora we use.

2.1 Introduction

For any computational processing of language we need evidence, and one such source of evidence is corpora. A corpus is a source of collected evidence for research on NLP and linguistic analysis. Based on two modes of a language, a corpus may be speech or text. The use of corpora has added a new dimension to the study of linguistics, blessed by computer technology, that enables one to prove and verify previously coined hypotheses and use linguistic evidence directly. Different schools based on different philosophies, define corpus in different ways. According to the "Cambridge Encyclopedia of the English language" [13] a corpus is

> "A large collection of linguistic data, either written texts or a transcription of recorded speech, which can be used as a starting point of linguistic description or as a means of verifying hypotheses about a language."

The development of a corpus is a cyclic process that consists of pilot survey, text/sample collection, empirical investigations of linguistic variation and again revision of the design [14]. As texts of a corpus are viewed as a sample of the language considered for analysis, one must be careful in selecting the text for a corpus. Thus, a corpus is well-planned, well-formatted and is generally designed to serve a specific purpose and a specific interest in language processing. This distinguishes a corpus from an electronic archive or a library. In general, a corpus does not contain new information, rather with the help of processing software we can draw conclusions (which exist but most likely, not explicit) from different perspectives, that are normally based on frequency and collocation [15].

Being an international language, there are a number of internationally known and popular corpora for English. Among these, the British National Corpus (BNC)[1], the American National Corpus[2], the London-Lund Corpus (LLC)[3], and the Brown corpus[4] are the main ones. A collaborative project, *Enabling Minority Language Engineering* (EMILLE) [5] developed corpora for South Asian languages. Although India is a country with diverse languages, other than EMILLE, there are only a few corpora that serves the linguistic community of Indian languages.

[1]http://www.natcorp.ox.ac.uk; Access date: 3-July-2013
[2]http://americannationalcorpus.org; Access date: 3-July-2013
[3]http://www.helsinki.fi/varieng/CoRD/corpora/LLC/index.html; Access date: 3-July-2013
[4]http://www.helsinki.fi/varieng/CoRD/corpora/BROWN/index.html; Access date: 3-July-2013
[5]http://www.lancs.ac.uk/fass/projects/corpus/emille; Access date: 3-July-2013

In this chapter, we discuss the preprocessing and development of Assamese text corpora. For any text corpus, newspapers and books are the most commonly used sources. Now-a-days, free resources on the Internet are widely used as the prime sources. A balanced corpus covers all the properties of a language and is a good representative of the language. For the experiments reported throughout the thesis, the input data is in Unicode. In the Unicode, the range U0980-U09FF is assigned for Assamese-Bengali script although there is confusion in the naming of the script by the Unicode Consortium. For our experiments, we use four different collections of raw text. The next sections describes the corpora.

- EMILLE/CIIL corpus
- Pratidin corpus
- Wikipedia corpus
- Tezu Assamese corpus

Table 2.1 gives the sizes of corpora (in terms of number of words) and the domain covered by each corpus.

Corpus	Total words	Domain covered
EMILLE	2,600,000	Literature, Science, Law, Reading material, Sports, History
Pratidin	300,000	News article
Wikipedia	450,800	Law, Science, Literature, Geography, Reading material, Sports, History, Eminent personality
Tezu Assamese	1,060,550	News article, Literature, Science, Medicine

Table 2.1: Statistics of the used/developed corpora.

2.2 Emille corpus

EMILLE[6] */ CIIL*[7] *corpus*; a Unicode encoded corpus of nearly 2.6 million words developed jointly by Lancaster University, UK and Central Institute of Indian Languages, India. The original corpus requires a lot of preprocessing, as it contains a large number of errors. The following are some types of errors that we have corrected programmatically and by extensive manual checking in the course of our experiments.

[6]Enabling Minority Language Engineering; http://lancs.ac.uk/fass/projects/corpus/emille
[7]Central Institute of Indian languages; http://www.ciil.org

- Bengali র (r) occurs in the corpus instead of Assamese ৰ (r).
- য় (j) occurs where য (z) should and vice versa.
- An unrecognised character occurs where Assamese ৱ (w) should.
- If second character of a conjunct is ব (b), then it disappears.
- There are many pattern-less spelling mistakes.

We have collected around 1,900 articles with 3,160 words per article in average. We find 221,880 unique words (inflected), among which 132,185 words occurs only once in the corpus. Table 2.2 gives the first fifteen most frequent words in the corpus.

2.3 Assamese Pratidin corpus

Pratidin corpus; an ASCII based news corpus of nearly 300,000 words collected from the online version of the Assamese daily *Asomiya Pratidin*. These are converted to a normalised ASCII based encoding, for viewing in non-Unicode compliant viewers. We have developed a converter to convert these ASCII encoded normalised text to UTF-8 format[8]. We find 38,060 unique words (inflected), among which 19,090 words occur only once in the corpus. Table 2.2 gives the first fifteen most frequent words in the corpus.

2.4 Wikipedia corpus

Wikipedia is a well-known resource of free and high quality text. Articles are written using the MediaWiki Markup Language that provides a simple notation for text formatting with mark-ups such as bold, italic, underline, image and table. We collected the Assamese Wikipedia dump from http://dumps.wikimedia.org. The dump is in form of XML files. WikiExtractor[9], a freely downloadable application is used to extract text from the dump. We have collected around 2,000 articles with 225 words per article on average. We find around 65,000 unique words (inflected), among which 37,560 words occur only once in the corpus. Table 2.2 gives the first fifteen most frequent words in the corpus.

[8]The converter is freely downloadable at http://www.tezu.ernet.in/~nlp/r2u.htm
[9]http://medialab.di.unipi.it/Project/SemaWiki/Tools/WikiExtractor.py

Sl.	Emille (F)	Pratidin (F)	Wikipedia (F)	Tezu (F)
1.	আৰু (46577)	আৰু (5356)	আৰু (10757)	আৰু (20241)
	(aru:*and*)	(aru:*and*)	(aru:*and*)	(aru:*and*)
2.	এই (22809)	পৰা (3543)	হয় (5857)	এই (10092)
	(ei:*this*)	(pɒra:*from*)	(hɒj:*be*)	(ei:*this*)
3.	কৰি (15916)	কৰি (2802)	এই (4522)	কৰি (8965)
	(kɒri:*to do+IF*)	(kɒri:*to do+IF*)	(ei:*this*)	(kɒri:*to do+IF*)
4.	হয় (13633)	এই (2706)	কৰে (4490)	সেই (7486)
	(hɒj:*to be*)	(ei:*this*)	(kɒre:*to do+3PPrT*)	(sei:*that*)
5.	পৰা (13541)	কৰা (2549)	কৰা (4036)	আছে (6822)
	(pɒra:*from*)	(kɒra:*to do+2PPrT*)	(kɒra:*to do+ 2PPrT*)	(ase:*to be+3PPrT*)
6.	কৰা (12922)	বাবে (1981)	পৰা (3004)	হয় (6187)
	(kɒra:*to do+2PPrT*)	(babe:*for*)	(pɒra:*from*)	(hɒj:*be*)
7.	এটা (12824)	যে (1892)	চনত (2705)	যে (5718)
	(eta:*one*)	(ze:*that*)	(sɒnɒt:*year+LM*)	(ze:*that*)
8.	হৈ (12462)	কৰে (1809)	আছিল (2367)	হৈ (5242)
	(hoi:*happen+IF*)	(kɒre:*to do+3PPrT*)	(asil:*to be+PT*)	(hoi:*happen+IF*)
9.	সেই (10483)	বুলি (1697)	তেওঁ (2312)	কৰা (4889)
	(sei:*that*)	(buli:*as if*)	(teo$\tilde{}$:*he*)	(kɒra:*to do+2PPrT*)
10.	কথা (10162)	হৈ (1653)	কৰি (2312)	বুলি (4454)
	(kɒt^{h}a:*speech*)	(hoi:*happen+IF*)	(kɒri:*to do+IF*)	(tar:*his*)
11.	কিন্তু (9804)	হৈছে (1566)	বাবে (2290)	এটা (4013)
	(kintu:*but*)	(hɒise:*to be+3PPrT*)	(babe:*for*)	(eta:*one*)
12.	আছে (9686)	কিন্তু (1544)	হৈছে (1918)	কিন্তু (3840)
	(ase:*to be+FF*)	(kintu:*but*)	(hɒise:*3PPrPT*)	(kintu:*but*)
13.	তাৰ (9496)	আজি (1428)	ইয়াৰ (1857)	নাই (3684)
	(tar:*his*)	(azi:*today*)	(iyar:*it's*)	(nai:*do not*)
14.	নাই (9469)	কৰিছে (1382)	কৰিছিল (1699)	বাবে (3410)
	(nai:*do not*)	(kɒrise:*to do+3PPrPT*)	(kɒrisil:*to do+3PPPT*)	(babe:*for*)
15.	বুলি (9188)	এক (1193)	চনৰ (1685)	কৰিছে (3225)
	(buli:*as if*)	(ek:*one*)	(sɒnɒr:*year+GM*)	(kɒrise:*to do+3PPrPT*)

F−Frequency; IF−Infinite form; GM−Genitive Marker; LM−Locative Marker;
2PPrT−Second Person Present Tense; 3PPrT−Third Person Present Tense;
3PPT−Third Person Past Tense; 3PPPT−Third Person Past Perfect Tense;
3PPrPT−Third Person Present Perfect Tense.

Table 2.2: First fifteen most frequent words from the corpora.

2.5 Tezu Assamese corpus

The *Tezu Assamese corpus* is a collection of web-blog, and news article collected from online Assamese news-papers and electronic magazines. The task of downloading web pages from web sites was automated by means of the standard UNIX tool wget. The tool

```
<article>
<desc>
<encoding>UTF-8</encoding>
<lang>Assamese</lang>
<pagesource>http://xondhan.com/%E0%A6%85%E0%A6%B6%E0%A6%BF%E0%A6%95%
E0%A7%8D%E0%A6%B7%E0%A6%BF%E0%A6%A4-%E0%A6%AE%E0%A6%BE%E0%A6%A8%
E0%A7%81%E0%A6%B9%E0%A7%B0-%E0%A6%B8%E0%A6%82%E0%A6%96%E0%A7%8D%
E0%A6%AF%E0%A6%BE%E0%A6%A7%E0%A6%BF</pagesource>
<date>15-07-2012:20.00</date>
</desc>
<body>
অসমত নিউজ চেনেলসমূহৰ ব্যৱসায়ীক উত্থান আৰু ৰাজনৈতিক পক্ষপাতিত্বমূলক বাতৰি প্ৰচাৰ সম্পৰ্কীয় বিতৰ্কই এটা কথা
প্ৰমাণ কৰিলে আৰু সাধাৰণ নাগৰিকৰ মনলৈকো এই স্পষ্ট ধাৰণা আনি দিলে যে- ৰাজনৈতিক ক্ষেত্ৰখনত এতিয়াও সংবাদ-
মাধ্যমে বিৰাট প্ৰভাৱ বিস্তাৰ কৰে; কাৰণ সেইটো নোহোৱা হ'লে পক্ষপাতদুষ্ট বাতৰি প্ৰচাৰ কৰাতকৈ মূল বাতৰিকে প্ৰচাৰ
কৰিবলৈ সকলোৱে সাহস কৰিলেহেঁতেন। ....
</body>
</article>
```

Figure 2.1: Structure of XML corpus in Tezu Assamese corpus.

gathers all the pages from a site as HTML pages. We develop a simple HTML parser that takes the Assamese text from a web page and stores the extracted Assamese text into a XML file. On providing a directory that contains the HTML pages, the parser generates an XML file for each HTML file. Figure 2.1 describes the structure of a XML file. We collect around 2,950 articles with 360 words per article on average. We find around 182,650 unique words (inflected), among which 87,150 words occurs only once in the corpus. Table 2.2 gives the first fifteen most frequent words in the corpus.

2.6 Summary

In this chapter, we describe two pre-processed corpora and two corpora we have developed in the course of our work. We found that throughout all the corpora developed and preprocessed, আৰু (aru : *and*) is the most frequent word. We also observed that among uniquely infected words 45-60% words in the corpus has single occurrence. Although the sizes of the corpora are small, it covers almost all possible domains including medicine. The next chapter will describe our attempt in finding word stems from corpora described in this chapter.

Chapter 3

Stemming in Assamese words

"When all the inflectional affixes are stripped from the words of a language, what is left is a stock of stems"

– A course in modern linguistics, C. F. Hoket (1916 - 2000)

Outline: This chapter presents the following:

1. A brief introduction to stemming.
2. A brief description about previous works related to stemming.
3. Stemming related issues of Assamese.
4. Description of approaches used to extract stem and analysis of result.
5. Discussion and concluding remarks.

3.1 Introduction

An information retrieval (IR) system attempts to identify and retrieve relevant information from a database, usually containing a large number of documents. A document in IR is usually represented as a set of words. The efficiency of an IR system is adversely affected by the abundance of words appearing in various morphological forms, either as a result of inflection or derivation. To reduce this adverse effect of morphological variation, one common method is to represent the words in a normalized representative form[1]. One approach to do so is by finding the root word from an inflected or a derivational form; this is known as *stemming*. It is an initial step in analyzing the morphology of words. Thus, instead of keeping all the variants in the database, if we store or index only the base word[2], it reduces the size of the index. Thus, our problem is to find the base form(s) of a given word or a set of words.

Methods for finding the base form include affix stripping, co-occurrence computation, dictionary look-up and probabilistic approaches. Most approaches are first developed for English, and later adapted for other languages (see Section 3.2). These approaches may not work properly for highly inflectional languages, including Indian languages which are our focus. These languages are morphologically rich and relatively free word order. Studies by McFadden [16, Chapter 5] and Müller [17] analyze the relationship between morphology and word order freedom of natural languages. A number of approaches for stemming have been used by researchers for Indian languages. Reported approaches can be classified into three broad categories: rule-based techniques [18, 19], supervised techniques [20, 21], and unsupervised techniques [22, 12, 23]. The rule-based approach develops rules based on linguistic analysis without training. Usually, the rule-based approach produces the best results for either relatively fixed word order or languages with a limited amount of inflection. However, highly inflectional languages need more in-depth linguistic knowledge of the formation of words to handle more complex derivational and inflectional morphology. In highly inflectional languages, it is very common to see compound formation, partial and full combination of two words, and abundant conflation of tense, aspect and mood markers to the root. Our objective is to reverse the effect of inflection or derivation on a stem. In other words, given a "complex" word, we want to

[1]In IR, there are two popular methods to generate the normalized representative form from a given word - stemming and lemmatization. Lemmatization is more useful where the manner of inflection is predominantly irregular. Common words in Assamese do not require lemmatizer. Inflections of a few Assamese verbs are irregular and with the help of a simple rule-base we determined the stem from the irregular verbs.

[2]Though the terms *base form*, *stem* and *root* differs linguistically, for this report we are using these three terms to mean stem.

find its morphological constituents, in particular identify its stem. A machine learning approach, whether supervised, unsupervised or rule-based, needs linguistic resources including substantial corpora, which are sorely lacking in resource-poor languages. Thus, a machine learning approach may not produce better results compared to hand-crafted rules for resource-poor languages like most Indian languages. The two main contributions of this chapter are the following.

- In this study, we take into consideration the problem of stemming Assamese (a resource-poor language from and North-east India) texts for which stemming is hard due to the morphological richness of the language. We use three different techniques to find the stem, explained step by step in the following sections. Our experiments reveal that approximately 50% of the inflections in Assamese appear as single letter suffixes. Such single letter morphological inflections cause ambiguity when one predicts the underlying root word.

- After obtaining encouraging result in Assamese (~16.5 million native speakers)[3], we use the approaches to stem text in several other Indian languages, viz, Bengali (~181 million native speakers)[4], Bishnupriya Manipuri (~1.15 million native speakers)[5] and Bodo (~1.54 million native speaker)[6] to show the level of generality in our method. Bishnupriya Manipuri and Bodo are vulnerable language according to UNESCO[7].

The rest of the chapter is organized as follows. We give a brief description of prior work related to stemming in Section 3.2, followed by the linguistic characteristics of Assamese. The next three sections describe the approach used for stemming. Each section contains results and discussion. Section 3.7 describes results obtained for the three additional Indian languages. Section 3.8 gives the concluding remarks. For transliteration of given examples, we use an in-house transliteration scheme and also provide representation using International Phonetics Alphabet (IPA).

[3]According to http://www.ethnologue.com; access date: 30 January 2013

[4]According to http://www.ethnologue.com; access date: 30 January 2013

[5]According to http://en.wikipedia.org/wiki/Bishnupriya_Manipuri_language; access date: 30 January 2013

[6]According to http://en.wikipedia.org/wiki/Bodo_language; access date: 30 January 2013

[7]http://www.unesco.org/culture/languages-atlas/index.php

3.2 Related work

In most well-studied languages, morphological inflections usually take place at the end of a word-form and this has influenced the affix stripping approaches to extract the root from a given word. The Porter stemmer [18], an iterative rule-based approach, has found the most success and is used widely in applications such as spell-checking, and morphological analysis. This simple approach was first developed for English and later adapted to Germanic (German, Dutch), Romance (Italian, French, Spanish and Portuguese) and Scandinavian languages (Swedish, Finnish, Danish and Norwegian), Irish, Czech, Armenian, Basque, Catalan, and Russian [24]. Lovins [25] introduced a suffix dictionary to assist in finding stems of words. The right-hand end of a word is checked for the presence of any of the suffixes in the dictionary. These two algorithms pre-date the development of many other algorithms such as [26, 27, 28]. Oard et al. [29] discover suffixes statistically using a four-stage backoff technique from a text collection and eliminate dependence on word ending. They count the frequency of every one, two, three, and four character suffixes (in decreasing order) that would result in a stem of three or more characters for the first 500000 words of the collection. A probabilistic stemming approach is described by Dincer and Karaoglan [30] for Turkish. String distance-based stemming, an alternative to language-specific stemming is proposed by Snajder and Basic [31], where stems are classified using a string distance measure called the dice coefficient based on character bigrams from a corpus. McNamee et al. [32] develop a system which combines word based and 6-gram based retrieval, performing remarkably well for several languages (English, French, German and Italian). One major pitfall with the n-gram approach is the increase in the size of the inverted index. A series of experiments were conducted by Kraaij and Pohlmann [33] to enhance the recall of stemming at the cost of some precision. They find that stemming of derivational words reduces precision by a considerably higher amount than inflectional words for Dutch.

European languages including English have a number of stemmers available and their performance has been extensively examined [27, 28]. Some other reported stemmers include French [34, 22], Spanish [22], Finnish [35], Czech [36] and Hungarian [23]. Arabic [37, 38, 39, 40], Japanese [41, 42], German [43, 32] and Dutch [33] are also well studied in the literature. For language-specific stemming, additional resources (like dictionary) are also often used [44] to group morphologically related words.

In the Indian language context, a few stemmers have been reported. Among these, Ramanathan and Rao [45] use a hand crafted suffix list and strip off the longest

suffixes for Hindi and report 88% accuracy using a dictionary of size 35,997. The work reported by Majumder et al. [22] learns suffix stripping rules from a corpus and uses a clustering-based method to find equivalent categories of root words. They show that their results are comparable to Porter's and Lovin's stemmers for Bengali and French. The work of Pandey and Siddiqui [46] focuses on heuristic rules for Hindi and report 89% accuracy. Aswani and Gaizauskas [47] propose a hybrid form of Majumder et al. [22] and Pandey and Siddiqui [46] for Hindi and Gujarati with precisions of 78% and 83%, respectively. Their approach takes both prefixes as well as suffixes into account. They use a dictionary and suffix replacement rules and claim that the approach is portable and fast. Sharma et al. [12] describe an unsupervised approach that learns morphology from an unannotated corpus and report 85% precision. They discuss salient issues in Assamese morphology where the presence of a large number of suffixal determiners, *sandhi, samas*, and the propensity to use suffix sequences make more than 50% of the words used in written and spoken text inflected. Paik and Parui [23] report a generic unsupervised stemming algorithm for Bengali and Marathi as well as Hungarian and English. Their approach is entirely corpus-based and does not employ language-specific rules. A graph-based stemming algorithm is proposed by Paik et al. [48] for information retrieval. They report their experiment with two Indian (Marathi and Bengali) and five European (Hungarian, Czech, English, French, and Bulgarian) languages. Reported work of Kumar and Rana [49] for Punjabi uses a dictionary of size 52,000 and obtain 81.27% accuracy using a brute-force approach. Majgaonker and Siddiqui [50] describe a hybrid method (rule-based + suffix stripping + statistical) for Marathi and claim 82.50% precision for their system. Work in Malayalam [51] uses a dictionary of size 3,000 and reports 90.5% accuracy using finite state machines. Among the reported works on Indian languages, the result may vary widely as each author may have individual rules and corpus for the same reported language. As the languages considered in this paper except Bengali, are among the most resource-poor languages in the world, we work with a rule-based and a supervised approach, rather than following the current trends towards corpus-based unsupervised stemming [35, 38, 29, 23, 48]. In the languages we work with, large labelled corpora simply do not exist.

3.3 Language related issues

Most Indian languages are studied infrequently in the global context. Among Indian languages Hindi, Bengali, Tamil and Telugu are studied more often. Other languages still lack even a single good corpus or basic language processing modules like stemmers and

morphological analysers that are freely available. In this work, we focus on Assamese, Bengali, Bishnupriya Manipuri and Bodo. The first three languages share the same writing convention and fall in the Eastern Indo-Iranian language group. Bodo, an important language of North-east India, is a member of the Tibeto-Burman language family, but uses Devanagari script for writing. There is no published work on morphological analysis of Bodo and Bishnupriya Manipuri. Some common properties of the languages under consideration are given below.

- All are relatively free word order. This means that the position of occurrence of a word within a sentence may change without changing the overall meaning. For complex sentences, phrases can change their position of occurrence within the sentence. Inside a phrasal/clausal boundary the sequence of word occurrence is normally fixed.
- The predominant word order is subject-object-verb (SOV) and more than one suffix can be attached to a root word sequentially. In comparison to suffixes, the number of prefixes is very small.
- They share a small common vocabulary, although we are not interested in measuring the number of common vocabulary items among the languages.
- All are classifier-based verb final languages, that is, verb changes with person and TAM (tense, aspect and modality) markers, not with gender and number.

Language	Ram	Killed	Ravana
English	ɾam	kɪl + d	ɾawɒnɒ
Assamese	ɾam + ɛ	maɾ + isil	ɾabɒnɒ + k
Bengali	ɾam	mɛɾ + ɛʧilo	ɾabɒn + kɛ
Bishnupriya Manipuri	ɾam+ɛ	maɾ + ɛʧil	ɾabɒn + ɒɾɛ
Bodo	ɾam + a	but^{h}aɾ + dɯɔŋmon	ɾaban + k^{h}ɔɯ
Hindi	ɾam nɛ	maɾa	ɾawɒn ko
Manipuri	ɾam nɔ	hatlami	ɾawɒn bu
Nepali	ɾam lɛ	maɾako t^{h}io	ɾawɒn lai
Oriya	ɾam	maɾi t^{h}ilɛ	ɾawɒn ku

Table 3.1: Pattern of inflections in some Indian language and English for the sentence *Ram killed Ravana* irrespective of the word position.

The manner in which morphology is expressed varies from language to language. For example, among Indian languages, for Hindi, Oriya, Manipuri and Nepali, most of

the time morphological attributes are separate tokens whereas in the case of Assamese, Bishnupriya Manipuri, Bodo and Bengali, the morphological attributes are always part of the words and thus need separate methods to handle. We present some common patterns of adding morphological inflection in Table 3.1. In the case of Assamese, Bodo and other similar languages, stemming is the process of finding sub-string(s) in a token. In the context of stemming, the most common property of languages like Assamese is that, they take a sequence of suffixes after the root words. We give some examples below.

1. **Assamese:**
   নাতিনীয়েককেইজনীমানেহে→নাতিনী + য়েক + কেইজনী + মান + ে + হে
   natinijɛkkɛizɒnimanɛhɛ→natini + jɛk + kɛizɒni + man + ɛ + hɛ
   nAtinIyekkeijanImAnehe → grand-daughter + inflected form of kinship noun[8] + indefinite feminine marker + plural marker + nominative case marker + emphatic marker

2. **Bengali:**
   ভেবেছিলাম → ভেবে + ছিল + াম
   b^hebeʧilam → b^hebɛ + ʧilo + am
   bhebesilAm → to think + past tense marker + person marker

3. **Bishnupriya Manipuri:**
   মানুকতোগইহে→ মানু + কতোগো + ই + হে
   manukɒtoguihɛ → manu + kɒtogu + i + hɛ
   mAnukatogoIhe→ man + indefinite plural marker + nominative case marker + emphatic marker

4. **Bodo:**
   রাজাফোরনিনোমোন → রাজা + ফোর + নি + নো + মোন
   razap^hərninəmən → raza + p^hər + ni + nə + mən
   rAjAfwrninwmwn → king + plural marker + genitive marker + definitive marker + remote tense marker

During literature survey we found that an Assamese noun root word may have 3,500-6,000 inflectional forms, although the maximum number of suffixes attached in sequence after the root seems to be limited to five. Among Indo-Aryan languages, Assamese

[8]In Assamese, the য়েক (jɛk) suffix is appended after all relational nouns in 3^{rd} person. For instance, the relational noun ভাই (**b^hai** : *younger brother*) is inflected to ভায়েক (**b^haijɛk**) and the relational noun ককাই (**kɒkai** : *elder brother*) is inflected to ককায়েক (**kɒkajɛk**). [52] reports that Assamese has the highest number of kinship nouns among Indo-Aryan languages.

has the largest number of relational nouns to denote relations between two persons [53]. A relational noun root in Assamese may have 10000-15000 inflectional forms depending on nominal attributes like number, gender, animacy and emphasis. In Appendix A (Page no.: 104), we list 50 inflectional forms of the noun root মানুহ (manuh : *man*). Likewise, an Assamese verb may also have 300-1500 different inflectional forms depending on person, tense, aspect, honor, mood and emphasis. In Appendix A (Page no.: 106), we tabulate some inflectional forms of the root আছ (aso : *to be*). Indian languages, including Assamese, have two types of vowels: one is the full vowel and another is the vowel "matra". During clubbing suffix sequences into a word, morpho-phonemic changes occur depending on the ending of the base word and the starting of the suffix to be appended. For example, the word আই (ai : *mother*) ends with the full vowel ই (i). Adding the nominative suffix এ (je) at the end of the root word আই (ai) results a new word আয়ে (aije). Although phonetically (see the sequence of IPA symbols) there is no change after addition of the new suffix, there is a change in the orthography. That is, after clubbing, the vowels ই (i) and এ (je) are changed to য়ে (je), semi-vowel + vowel matra. The next section discusses a rule-based approach to stemming, focusing on Assamese.

3.4 Approach-1: Rule-based approach

As mentioned above, an Assamese root can take a series of suffixes sequentially. So, our first aim is to find the probable sequence of suffixes that a word contains. We manually collected all possible suffixes and categorized them into six basic groups, viz., case marker (CM) (nominative, accusative, locative, genitive, instrumental and dative), plural marker (PM), definiteness marker (DM), emphatic marker (EM), verb markers (VM) and others. The other categories contain kinship noun markers, adverbial makers and derivational suffixes. Table 3.2 shows some suffixes with their counts.

Case	Plural	Definiteness	Verb	Emphatic
-এ (e)	-বোৰ (bor)	-জন (ʤɒn)	-ই (i)	-চোন (ʃon)
-ক (k)	-হঁত (ɦɒt)	-গৰাকী (gɒraki)	-া (a)	-নে (ne)
-ৰে (re)	-সমূহ (xɒmuh)	-খন ($k^hɒn$)	-ইছিলা (isila)	-ও (o)
-লৈ (loi)	-মখা ($mɒk^ha$)	-টো (to)	-ইবি (ibi)	-হে (he)
-ৰ (r)	-থোপা (t^hopa)	-পাত (pat)	-ইলা (ila)	
-ত (t)	-মান (man)	-কেইজনী (keiʤoni)	-োৱাইছিলোঁ (ojaisilo)	
Total: 7	72	98	134	4

Table 3.2: Categories of suffixes with examples

The **rule engine** generates a list of suffixes. It is a module that generates all probable suffix sequences that may be attached to a root, based on the affixation rules incorporated in the engine. It uses a collection of rules and produces a valid list of suffixes in proper sequence. By proper sequence, we mean that the suffixes must abide by morphotactic rules for Assamese. For example, the addition of a plural marker after a case marker will generate an invalid sequence for Assamese. We have observed that no inflections are attached to the verb root or noun root after the emphatic marker. The following examples illustrate such affixation rules that the rule engine uses to generate the suffix list. The first eleven, Examples (5-15), illustrate nominal suffixation. The word মানুহ (**manuh** : *man*) is the root in Examples 5 through 15 and the word কৰ (**kɒr** : *to do*) is the root in Examples 16 and 17.

5. root + PM
   Example: মানুহবোৰ → মানুহ + বোৰ
   IPA: manuhbɒr → manuh (*man*) + bɒr (*plural marker*).

6. root + CM
   Example: মানুহৰ → মানুহ + ৰ
   IPA: manuhɒr → manuh (*man*)+ ɒr (*genitive case marker*).

7. root + DM
   Example: মানুহজন → মানুহ + জন
   IPA: manuhɒr → manuh (*man*)+ zɒn (*definitive marker*).

8. root + EM
   Example: মানুহহে → মানুহ + হে
   IPA: manuhhe → manuh (*man*)+ he (*emphatic marker*).

9. root + PM + CM
   Example: মানুহবোৰৰ → মানুহ + বোৰ + ৰ
   IPA: manuhbɒrɒr → manuh (*man*) + bɒr (*plural marker*)+ ɒr (*genitive case marker*).

10. root + PM + EM
    Example: মানুহবোৰহে → মানুহ + বোৰ + হে
    IPA: manuhbɒrhe → manuh (*man*) + bɒr (*plural marker*)+ he (*emphetic marker*).

11. root + CM + EM
    Example: মানুহৰহে → মানুহ + ৰ + হে
    IPA: manuhɒrhe → manuh (*man*) + ɒr (*genitive case marker*)+ he (*emphatic marker*).

12. root + DM + CM
    Example: মানুহজনৰ → মানুহ + জন + ৰ
    IPA: manuhzɒnɒr → manuh (*man*) + zɒn (*definitive marker*)+ ɒr (*genitive case marker*).

13. root + DM + EM
    Example: মানুহজনহে → মানুহ + জন + হে
    IPA: manuhzɒnhe → manuh (*man*) + zɒn (*definitive marker*)+ he (*emphatic marker*).

14. root + PM + CM + EM
    Example: মানুহবোৰকহে → মানুহ + বোৰ + ক + হে
    IPA: manuhbɒrɒkhɛ → manuh (*man*) + bɒr (*plural marker*)
    + ɒk (*accusative case marker*) + hɛ (*emphatic marker*).

15. root + DM + CM + EM
    Example: মানুহজনকহে → মানুহ + জন + ক + হে
    IPA: manuhzɒnɒkhɛ → manuh (*man*) + zɒn (*definitive marker*)
    + ɒk (*accusative case marker*) + hɛ (*emphatic marker*).

16. root + VM
    Example: কৰিছিলোঁ → কৰ + িছিলোঁ
    IPA: kɒrisilõ → kɒr (*to do*) + isilõ (*past tense marker, 1st person*).

17. root + VM + EM
    Example: কৰিছিলোঁনে → কৰ + িছিলোঁ + নে
    IPA: kɒrisilõnɛ → kɒr (*to do*) + isilõ (*past tense marker, 1st person*)
    + nɛ (*emphatic marker*).

18. root + others
    Example: যাওঁহক → যা + ওঁ + হক
    IPA: zaõhɒk → za (*to go*) + õ (*1st person marker*) + hɒk (*other suffix*)

We confirmed 14 (Examples 5-18) such rules and generated 18194 suffix sequences for Assamese. These rules are implemented using Java RE package. Examples 9 through 15 and Examples 17 and 18 have sequences of two or more suffixes whereas Examples 5 through 8 and 16 have only one suffix each, attached to the root. The suffix list generated by the rule engine contains the suffixes বোৰ (bɒr), ৰ (r) and জনকহে (zɒnɒkɒhe) in Examples 5, 6 and 15, respectively. The rule engine works with all grammatical categories. It generates a list of suffix sequences in non-increasing order of the length of the sequence, so that in the next phase the longest possible suffix sequence in an input

word can be identified using a sequential look-up of the list. For instance, let us consider the word নাতিনীয়েককেইজনীমানেহে (natinijɛkkɛizɒnimanɛhɛ) (Example 1) and assume the suffix sequence list has the entries হে (hɛ), মানেহে (manɛhɛ), জনীমানেহে (zɒnimanɛhɛ), কেইজনীমানেহে (kɛizɒnimanɛhɛ) and য়েককেইজনীমানেহে (jɛkkɛizɒnimanɛhɛ) in that order. A sequential look-up will yield the segmentation natinijɛkkɛizɒnimanɛ + hɛ, whereas the segmentation natini + jɛkkɛizɒnimanɛhɛ is more appropriate. Hence the list is arranged in non-increasing order of length of the suffix sequences. The input words are passed through the suffix look-up process. For an input word of length L_w, matching is attempted only with the suffix sequences whose length is less than $L_w - 1$. Any match found in the list, will separate the word matched part as a component.

3.4.1 Results and discussion

In this experiment, we use a part of the EMILLE[9] Assamese corpus of size 123,753 words. Among these, 25,111 words are unique (including inflected and root words). For example, মানুহ (manuh : *man*), মানুহবোৰ (manuhbor : *men*) and মানুহটো (manuhto : *the man*) are considered separate words, although the second and the third words are inflected forms of the first word. We found that 5.85 is the mean word length for the corpus. We manually evaluate the output. One highly educated and native speaker is employed as manual evaluator and found 57% of the words are correctly stemmed by Approach 1. Some observations from these experiments that explain the low accuracy are enumerated below.

A. Suffixes such as -বোৰ (-bor), -মান (-man) and -জন (-zɒn) were separated from words such as কেতবোৰ (ketbor : *some*), কিছুমান (kisuman : *some*) and প্ৰয়োজন (prɒjozɒn : *need*) although these words are not inflected. Table 3.3 shows four types of errors found using the approach. As defined in Table 3.3, Type-I and Type-II errors are similar in the sense that they are about single suffixes, but the first one is about single letter suffixes and the second one is about multi-letter suffixes. Type-III and Type-IV are due to the merging of two suffixes. Although the sequence is correct, the error is in the identification of suffix boundaries.

B. The error rate for inflected words with suffix length greater than 4 is less than 1% whereas the error rate for inflected words with suffix length equal to 1 is the highest, 56%.

[9]http://www.emille.lancs.ac.uk/

Input word	Generated stem	Correct stem	Type of error	%
কাপোৰ (kapor)	কাপো (kapo)	কাপোৰ (kapor)	**Type-I error:** Found a single letter suffix at the end of the input word [ৰ (r), in the example] and removed [as genitive case marker, in the example] although the word is not inflected.	51
প্রয়োজন (prɒjozɒn)	প্রয়ো (prɒjo)	প্রয়োজন (prɒjozɒn)	**Type-II error:** Found a multi-letter suffix [জন (zɒn), in the example] at the end of the input word and removed [as definitive marker (zɒn) is in the suffix list]. Here the input word is a root word.	30
প্রয়োজনত (prɒjozɒnɒt)	প্রয়ো (prɒjo)	প্রয়োজন (prɒjozɒn)	**Type-III error:** Found suffix sequence [জন + ত (zɒn +t), in the example] generated by the rule engine and removed. Whether only the last letter ত (t) is the inflectional part in the input word, জন (zɒn) is not.	17
ডাঙৰে-ডাঙৰে (daŋɒre-daŋɒre)	ডাঙৰে-ডাঙ (daŋɒre-daŋ)	ডাঙৰ-ডাঙৰ (daŋɒr-daŋɒr)	**Type-IV error:** Input is a reduplicative (a common phenomenon [54] in Indian languages) word with a hyphen. The stemmer found suffix sequence ৰে (re) at the end of the second part of the reduplicative word and removed it. But the first part is also inflected and therefore, should be stemmed as well.	2

Table 3.3: Analysis of error types of Approach 1.

It is clear from this experiment that the error rate decreases with increasing suffix length. As the error rate of single character suffixes is the highest, one possible solution to increase the accuracy of stem identification is to add a root word list, which is discussed in Section 3.5.

3.5 Approach-2: Dictionary look-up-based approach

It is clear that using the rule set developed in the previous section, we were not able to extract all stems from the input words. On looking closely at the Type-I and Type-II errors, we find that the inputs are root words, but the end letter(s) unfortunately match some valid suffixes from the suffix list. In Table 3.3, we see that the input word কাপোৰ that causes a Type-I error, is a root word itself. The end letter of the input word, ৰ (r) is the genitive case marker and is in the suffix list. Hence, the stemmer separates ৰ (r) from কাপোৰ, producing a wrong stem কাপো. Similarly, our algorithm removes ৰ (r) as a suffix from all words that end in ৰ even though many such words are indivisible. A Type-II error is similar to a Type-I error, except the number of letters present in the suffix. The number of words that cause Type-II errors is fewer but such words occur frequently in the text. To handle these two types of exceptions, one way is to maintain a word list (henceforth, called *dictionary*) where most frequent stems or roots are kept. For example, words ending with any character listed in Table 3.5, such as ভাত (b^{h}at : *rice*), মাত (mat : *voice*), অমৰ (ɒmɒr)) and exceptional root words (such as কেতবোৰ (kɛtbor : *some*), কিছুমান (kisuman : *some*) and প্ৰয়োজন (prɒjozɒn : *need*)) are stored in a text file, one word per line. Thus in this approach, first each word is checked against the dictionary (that is, words stored in the text file) to be stemmed. After that we apply Approach 1. The main advantage of the approach is that it minimizes over-stemming (removing too many letters as suffix) and under-stemming (removing fewer letters as suffix) errors [c.f. Table 3.11].

3.5.1 Preparation of dictionary

We develop a frequent root word list from the entire EMILLE Assamese corpus (approximately 2.6 million words). Alternatively, the dictionary may comprise only those words that clash with the suffixes, thus may improve the search efficiency. Using a Python program, we extract the unique words (including inflected and root words) and their frequencies from the corpus in lexicographic order to ease the identification of the roots. We manually extract and arrange the root words based on frequency. Figure 3.1 illustrates our experiment to visualize the impact of the size of dictionary coverage in stemming. We choose 5000, 10000, 15000, 20000 and 25000 most frequent root words and test with the corpus described in Section 3.4.1. We found accuracies of 66, 73, 77, 80 and 81% respectively, when merged with Approach 1. This shows stemming accuracy increases as the size of the dictionary increases, although as expected the increase starts to level off. We also examine the accuracy (i.e., the number of root words) of stemming without

combining with Approach 1.

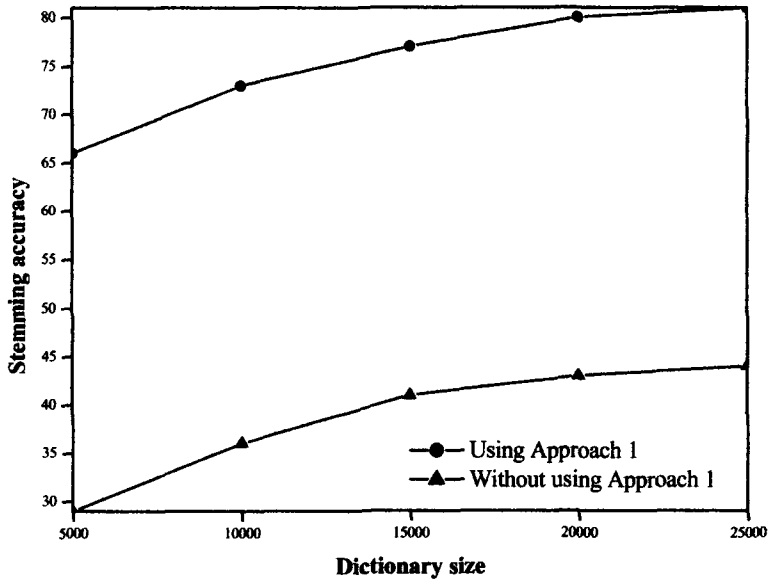


Figure 3.1: Impact of accuracy with increasing dictionary size

3.5.2 Results and discussion

For this approach we use a word list of size 25000 root words. As obvious, the obtained results in Table 3.4 are an improvement over the first approach. We use a set of hand-

Correctly stemmed	81%
Dictionary words	45%
Incorrectly stemmed	19%
Stemmed as no inflection	23%
Stemmed as single character inflection	57%
Stemmed as multiple inflection	20%

Table 3.4: Result obtained using dictionary with Approach 2.

crafted rules as discussed earlier and a dictionary as discussed in this section to stem and obtain an accuracy of nearly 81%. Nearly 19% of the words are still not stemmed properly. Among the incorrectly stemmed words, 23% of the words are marked root words although they possess inflection. That is, the rules fail to extract inflection from such words. On looking closely at the incorrectly stemmed words that are marked as no inflection, we

find these are mostly single character inflections attached to the root word. Of the words that are incorrectly stemmed, 57% are incorrectly stemmed as single character inflection. Digging deeper, we find that these words are not in the dictionary and most of them are proper nouns whose end letters are unfortunately the same as some single letter suffix. The common appearance of single letter suffixes as morphological inflections causes the rapid downfall of the accuracy in Approach 2. We find that, among the generated suffixes, 11 suffixes are single letter suffixes and more than 50% of the inflections in Assamese are

Suffix	Category	Inflected words	Root words
ক (k)	Acquisitive marker / 2^{nd} person present tense marker	ৰামক (ramɔk = ram + k)	কাৰক (karɔk)
ত (t)	Locative marker	কামত (kamɔt = kam + t)	ক'ত (kɔt)
ৰ (r)	Genitive marker	তেখেতৰ (tekʰetɔr = tekʰet + r)	অমৰ (ɔmɔr)
ি/ই (i)	Present participle / Nominative marker	কৰি (kɔri = kɔr + i)	কলি (kɔli)
া/আ (a)	Finite verb marker	কৰা (kɔra = kɔr + a)	কলা (kɔla)
ে/এ (e)	Finite verb marker / Nominative marker	নকৰে (nɔkɔre = n + kɔr + e)	দে (de)
ো/ও (o)	Finite verb marker	কৰো (kɔro = kɔr + o)	কোনো (kono)

Table 3.5: Root words whose final letters match some suffix.

single letter suffixes. Such single letter morphological inflections cause ambiguity while predicting the underlying root words. This approach eliminates the Type-II errors to a great extent and a fraction of the Type-I errors. A fraction of the Type-I errors still exists, particularly when the stemmer finds an unseen word that ends with a member of the single letter suffix set. Keeping this in mind our goal next is to further improve proper stemming of unseen words and increase accuracy. We describe a Hidden Markov Model in Section 3.6 to handle single letter inflections left out by the previous two approaches.

Language	Sente-nces	Words		Inflection type			Source of text
		Total	Unique	Single	MS	Multiple	
English	82	2012	843	06.88%	00.00%[10]	18.50%	Times of India[11]
Assamese	132	2164	1293	28.21%	09.49%	13.06%	Dainik Janasadharan[12]
Bengali	202	2205	1246	17.97%	07.22%	18.37%	Anandabazar Patrika[13]
Hindi	116	2162	795	12.07%	03.14%	12.82%	Dainik Jagaran[14]

Table 3.6: A random survey on occurrence of single letter suffix, multiple suffix and multiple suffix end-with single letter suffix (MS).

[10]Although English has words that end with suffix sequence, we do not find a single word in the randomly picked text.

3.6 Approach-3: A Hybrid Approach

Table 3.6 makes an important observation when we look at randomly picked news articles in English, Assamese, Bengali and Hindi. Each collection is approximately 2000 words. Among major Indian languages, Bengali is closest to Assamese in terms of spoken and written forms. Hindi is a closely related language as well, but written using a different script, the Devanagari Script. The fourth column gives the number of unique inflected words. We observe that among these languages, Assamese has the highest frequency of single letter inflectional suffixes. This behooves us to develop an algorithm to improve the accuracy of detecting single letter suffixes to build a better stemmer for Assamese. Melucci and Orio [55] use HMM for stemming five different languages, viz., Dutch, English, French, Italian and Spanish. They design their approach to stemming in terms of an HMM with states for two sub-processes or disjoint sets: states in the prefix-set that are considered to be the stems and states in the suffix-set that possibly generate the suffix sequence if the word has one. Our problem is a bit different. We intend to devise a model to learn to classify single letter suffixes only. Our work is first of its kind for some of the considered languages. Use of suffix is governed by syntactic principles of a language that may spread over an entire sentence. Since HMM is well known for sequence labelling, it is a suitable candidate for experiments like ours.

Our concept is very simple. We drop the occurrence of the single letter suffix set from the suffix list generated by the rule engine. We collect all the words, whose end character matches a member of the single letter suffix set, independent of inflectional information. This collection contains only those words, which are not in the dictionary and words that are not covered by the rule engine. This word list is sent as input to the HMM training model to classify. The task described here is an extension of our previous work [56].

3.6.1 The HMM model

Suppose $w_0, w_1, \cdots, w_{n-1}$ are the words of a corpus. Each word w_i can be split as $p_i \circ s_i$, where p_i is a root word; s_i an inflectional or derivational suffix; and $\circ$ the concatenation operation between two strings. Let S be the set of inflectional suffixes in the language

[11] http://timesofindia.indiatimes.com; *access date : 22-Nov-2012*
[12] http://janasadharan.in; *access date : 22-Nov-2012*
[13] http://www.anandabazar.com; *access date : 22-Nov-2012*
[14] http://www.jagran.com; *access date : 23-Nov-2012*

under consideration including the empty string ϵ. For any word, $w \equiv p \circ s$ if $s = \epsilon$, we say that word w is a *root word*, otherwise we say that word w ends with an inflectional or derivational suffix. Using this notation, the word আম (am : *mango*) can be decomposed as আম (am : *mango*) $\circ$ ϵ, as the end letter ম $\notin S$. The word মানুহৰ (manuhɔr : *of man*) can be represented as $p \circ s$ with $p =$ মানুহ (manuh : *man*) and $s =$ ৰ (ɔr) $\in S$. The word মানুহৰ (manuhɔr) is morphologically inflected. On the other hand, অমৰ (ɔmɔr : *immortal*) has $p =$ অম (ɔm) and $s =$ ৰ (ɔr). Although $s \in S$, অমৰ (ɔmɔr : *immortal*) is a root word. Thus, if there is an inflection $s \in S$ and $s \neq \epsilon$ such that $w = p \circ s$, we say w is morphologically inflected whether the generation is meaningful or not. Therefore, we define two states of the generator, G at the time of generating the word, *viz.*, inflected word (M) and root words or non-inflectional words (N). We can associate with a corpus of some length ℓ : $w_0, w_1, \cdots w_{\ell-1}$ a series of states, N and M's as $q_0, q_1, \cdots, q_{\ell-1}$ such that $q_i \in Q \equiv \{N, M\}$. For example in Table 3.7, we show the series of states for the sentence given in Example 19.

Example 19: নবীনহঁতৰ ঘৰ আমাৰ ঘৰৰ পৰা এমাইলমান দুৰত।

IPA: nɒbinɦɪtɒr $g^hɒr$ amar $g^hɒrɒr$ pɔra æmailman durɔt.

WWT[15]: nabin's(plural) house our house from one-mile distance.

AET[16]: The house of Nabin and his family is a mile from our house.

w	w_0	w_1	w_2	w_3	w_4	w_5	w_6
words	নবীনহঁতৰ	ঘৰ	আমাৰ	ঘৰৰ	পৰা	এমাইলমান	দুৰত
	nɒbinɦɪtɒr	$g^hɒr$	amar	$g^hɒrɒr$	pɔra	æmailman	durɔt
p	নবীন	ঘৰ	আমাৰ	ঘৰ	পৰা	এমাইল	দুৰ
	nɒbin	$g^hɒr$	amar	$g^hɒr$	pɔra	æmail	dur
s	-হঁতৰ	ϵ	ϵ	-ৰ	ϵ	-মান	-ত
	-ɦɪtɒr			-r		-man	-t
q	M	N	N	M	N	M	M

Table 3.7: An example sentence modelled using our generative model of the text for morphological inflections.

Therefore, for a corpus generated by G, the problem of deciding if a word is morphologically inflected boils down to determining the state of G (N or M) at the exact moment of generating the word. We construct an HMM-based algorithm to predict the states of G corresponding to the words of any given corpus. Therefore, the problem has two main aspects: (a) estimating the HMM parameters with a training corpus and (b) applying the calibrated algorithm on a test corpus to detect morphologically inflected

[15]WWT: Word to word Translation

[16]AET: Approximate English Translation

words. Our HMM for the generator G can be defined as follows.

i S is the alphabet that consists of the set of inflections that generator G can generate.

ii $Q = \{N, M\}$ is the set of possible states of G.

iii $A = (a_{kl})$ is the $|Q| \times |Q|$ matrix of state transition probabilities of G.

iv $E = (e_k(s))$ is the $|Q| \times |S|$ matrix that contains the emission probabilities of inflections (or the alphabet in the HMM) from S.

In order to compute the optimal path, we use the Viterbi algorithm [57]. The goal of the algorithm is to compute the probability $f_k(i)$ of the most probable path ending in state k at w_i for every possible state k. In our case, the states are either N or M.

We know that the inaccuracy of the previous method comes mostly from single letter inflections. For multiple letter inflections, the ambiguity of being a true inflection versus a coincidental match of the word with the set of inflections is significantly low. We denote by S_1 and S_m the set of single letter and multi-letter inflections, respectively. In order to simplify our analysis, we partition the set of inflections S as $\{\epsilon\}$, S_1 and S_m. Therefore, the appearance of a multi-inflection suffix on a word definitely generates the presence of morphological inflection. Hence, we can safely assume that if $s_i \in S_m$ for a word w_i, $q_i = M$. We can state the same notion as for $q_i = N$, $e_{q_i}(s) = 0$ for $s \in S_m$. Since we are essentially trying to predict the correct state of G for only single letter inflections (i.e., S_1) we assume all inflections in S_1 are equivalent and, similarity the inflections in S_m are equivalent to each other. So we assume that our alphabet S in the Hidden Markov Model is $S' = \{\epsilon, s_1, s_m\}$, where s_1 and s_m are single letter and multi-letter morphological inflections, respectively.

3.6.2 Preparation of training data

For this experiment, we used a random text from the EMILLE Assamese corpus. We labelled approximately 3,082 words with 4 tags.

- Words with multi-letter inflection (M_{sm}). For example: লিখিবলৈ (likʰibɒlɔi = likʰ + ib + lɔi : *to write* + future tense marker).
- Words with single character inflection (M_{s1}). For example: বয়সৰ (bɒjɒxɒr = bɒjɒx + r : *age* + genitive marker).

- Words with no inflection, i.e., root words (N_e). For example: সফল (xɒpʰɒl : *successful*).

- Words that have no inflection but end with a member of the single letter suffix set (N_{s1}). These are root words that end with a single letter suffix set. For example: যাৰ (ʤɒr : *whose*).

Table 3.8 gives details of suffixes present in the training set. It is clear from Table 3.8 that the number of words with single letter inflection (30.37% in the training data) is more than the number of words with multi-letter inflection (15.06% in the training data). Interestingly as mentioned earlier, words with single letter suffix (M_{s1}) and words that unfortunately end with any member of the single letter suffix set (N_ϵ) create problems. In the training set, we have 1686 (936+750) such words, which is more than 50% (54.67%) of the training data and 1682 words ($N_\epsilon + N_{s1}$) are actual non-inflected words. In addition

	Total	%
Words with single letter suffixes (M_{s1})	936	30.37
Words with multi-letter suffixes (M_{sm})	464	15.06
Words with no suffix (N_ϵ)	932	30.24
Root words that end with any member of the single letter suffix set (N_{s1})	750	24.33

Table 3.8: Suffix information in the Assamese training corpus with 3082 words

Suffix	Suffix category	Total	F (%)
ৰ (r)	Genitive marker	596	19.33
ত (t)	Locative marker	238	07.72
ক (k)	Acquisitive marker/2^{nd} person present tense marker	88	02.85
ি/ই (i)	Nominative marker/Non-finite verb marker	366	11.87
া/আ (a)	Finite verb marker	300	09.73
ে/এ (e)	Nominative marker/ Finite verb marker	276	08.95
োা/ও (o)	Finite verb marker	32	01.38

Table 3.9: Single letter suffix frequency (F) in the Assamese training corpus.

to 1686 (936+750) words that possess any one member of the single letter suffix set at the end, we have 210 more words in the training file, with multiple suffixes that end with any one member of the single letter suffix set. Using Approach 1 and Approach 2, we can handle multiple character inflection well. The statistics of single character inflection in the training set is given in Table 3.9. From the statistics, it is clear that the *genitive case marker* (19.33%) is the most frequent among the single letter suffixes. In Assamese, we have 11 suffixes in the single letter suffix set. The last four suffix pairs in Table 3.9 show

the *vowel matra* and the *full vowel*. Depending on use, they change their form from full vowel to matra or matra to full vowel. As shown in Table 3.9, among the 11 characters, ক (k), ই / ি (i) and এ/ে (e) are ambiguous, that is the same symbol/letter is used to inflect nouns as well as verbs.

3.6.3 Results and discussion

The stemming accuracy found using Approach 1 is 57% and using Approach 2 is 82% and using the hybrid approach is 94%. The complete statistics for our experiments are given in Table 3.10. Sharma et al. [12] reported 69% F-measure for suffix acquisition, when they tested their unsupervised approach with 300000 Assamese words. In comparison with Sharma et al [12], the result produced by Approach 2 with a root word list of only 25000 entries is considerably better. With the same test data, we found 85% correct stems using Morfessor [58]. Morfessor is an unsupervised language independent tool of four morphology learning models based on recursive minimum description length [59]. It takes an unannotated corpus as input and generates morpheme-like units of the words observed in the corpus. The remaining errors in our combined approach are due to irregular use of some verbs. As mentioned earlier, Assamese verb morphology is complex. Finding the root form from inflected forms of irregular verbs like বল (bɔl : *to go*) or যা (za : *to go*) is not possible with this stemmer. It needs a lemmatizer to extract the root, as the whole form of the irregular verb changes after inflection. However, with our hybrid approach we minimize the error rate to <10%. Embedding more linguistic knowledge using quantitative restrictions like [18, 25] and by proper handling of compound words and hyphenated words, we may be able to further reduce the error rate. In order to give

	Approach 1	Approach 2	Approach 3	Morfessor
Correctly stemmed	57%	81%	94%	81%
Dictionary words	-	45%	45%	-
Incorrectly stemmed	43%	19%	6%	15%
Stemmed as no inflection	24%	23%	36%	29%
Stemmed as single character inflection	56%	57%	33%	19%
Stemmed as multiple inflection	20%	20%	31%	52%

Table 3.10: Result obtained using various approaches.

a glimpse of the output difference among approaches, we tabulate the stem extracted by the various stemmers in Table 3.11 for some words containing the noun root নাটক (natɔk : *drama*). We consider the word নাটক (natɔk) because its last letter is a member of single letter suffix set. As obvious, in Approach 1, ক is extracted from the end of the root

word নাটক (natɒk) producing a wrong stem নাট (nat), although the extracted stem নাট is a valid meaningful word in Assamese. After introducing the word list in Approach 2, the stemmer recognizes নাটক (natɒk) as a root word. Morfessor, an unsupervised model, reports নাটক (natɒk) and নাটকখন (natɒkkʰɒn : *the darma*) as root words. In the second example, the extracted output নাটকখন (natɒkkʰɒn) is not a root word, whereas in the first example it is a root word. The authors of Morfessor state their approach as - "The general idea behind the Morfessor model is to discover as compact a description of the data as possible. Sub-strings occurring frequently enough in several different word forms are proposed as morphs and the words are then represented as a concatenation of morphs" [58]. Morfessor does not use language-specific rules. Based on evidence and probability, it learns to segment words into valid meaning bearing units. Therefore, in case of Examples (iv) through (vii) (Table 3.11), Morfessor produces the wrong stem as it finds নাটকখন (natɒkkʰɒn) to be a base word in the corpus. Due to the controlled rule sequence, our Approach 2 and Approach 3 produce the correct stem. Likewise, we compare the output of the stemmers in the Table 3.12 for some words containing the verb root কৰ (kɒr : *to do*).

3.7 Experiments in other languages

After obtaining excellent results in Assamese, we extend our approach to three other languages from Eastern India. Being spoken in the same region they partially share a vocabulary. For each language, we generated the suffix list using the rule engine and manually developed root word lists. We manually tagged 3212, 2540 and 2621 words using the four tags mentioned earlier for Bengali, Bishnupriya Manipuri and Bodo respectively and trained them for the hybrid model.

1. Like Assamese, the Bengali verb is a complex category in terms of inflection. Finite verbs in Bengali are inflected for person, tense, aspect, honor, mood and emphasis [19]. Unlike the verb, the base form is not changed in Bengali noun inflections. Unlike most other Indian languages, there have been several attempts at stemming Bengali texts [22, 19, 23, 48]. However, none of these stemmers are publicly available. After manual validation of the rules generated by the rule engine, we found 12456 suffix sequences using 11 rules. The following are the rules used in rule engine.

   (a) root + PM

Sl.	Word	Stemmer			Morfessor	Correct stemming
		Approach 1	Approach 2	Approach 3		
(i)	নাটক	নাট+ক	নাটক	নাটক	নাটক	নাটক (*drama*)
	(natɒk)	(nat + k)	(natɒk)	(natɒk)	(natɒk)	(natɒk)
(ii)	নাটকত	নাটক+ত	নাটক+ত	নাটক+ত	নাটক+ত	নাটক+ত (*drama+LCM*)
	(natɒkɒt)	(natɒk + t)	(natɒk + t)	(natɒk + t)	(natɒk + t)	(natɒk + t)
(iii)	নাটকক	নাটক+ক	নাটক+ক	নাটক+ক	নাটক+ক	নাটক+ক (*drama+ACM*)
	(natɒkɒk)	(natɒk + k)	(natɒk + k)	(natɒk + k)	(natɒk + k)	(natɒk + k)
(iv)	নাটকখন	নাটক+খন	নাটক+খন	নাটক+খন	নাটকখন	নাটক+খন (*drama+DM*)
	(natɒkkʰɒn)	(natɒk + kʰɒn)	(natɒk + kʰɒn)	(natɒk + kʰɒn)	(natɒkkʰɒn)	(natɒk + kʰɒn)
(v)	নাটকখনে	নাটক+খনে	নাটক+খনে	নাটক+খনে	নাটকখন + ে	নাটক +খন+ে (*drama+DM+NCM*)
	(natɒkkʰɒnɛ)	(natɒk + kʰɒnɛ)	(natɒk + kʰɒnɛ)	(natɒkkʰɒn + ɛ)	(natɒk + kʰɒnɛ)	(natɒk + kʰɒn + ɛ)
(vi)	নাটকখনি	নাটক+খনি	নাটক+খনি	নাটক+খনি	নাটকখন + ি	নাটক+খনি (*drama+DM*)
	(natɒkkʰɒni)	(natɒk + kʰɒni)	(natɒk + kʰɒnɪ)	(natɒk + kʰɒni)	(natɒkkʰɒn + ɪ)	(natɒk + kʰɒni)
(vii)	নাটকখনৰ	নাটক+খনৰ	নাটক+খনৰ	নাটক+খনৰ	নাটকখন+ৰ	নাটক+খন+ৰ (*drama+DM+GCM*)
	(natɒkkʰɒnɒr)	(natɒk + kʰɒnr)	(natɒk + kʰɒnɒr)	(natɒk + kʰɒnɒr)	(natɒkkʰɒn + r)	(natɒk + kʰɒn + ɒr)
(viii)	নাটকসমূহৰ	নাটক+সমূহৰ	নাটক+সমূহৰ	নাটক+সমূহৰ	নাটক+সমূহৰ	নাটক+সমূহ+ৰ (*drama+PL+GCM*)
	(natɒkxɒmuhɒr)	(natɒk + xɒmuhɒr)	(natɒk + xɒmuhɒr)	(natɒk + xɒmuhɒr)	(natɒk + xɒmuhɒr)	(natɒk + xɒmuh + ɒr)
(ix)	নাটকবোৰত	নাটক+বোৰত	নাটক+বোৰত	নাটক+বোৰত	নাটক+বোৰত	নাটক+বোৰ+ত (*drama+PL+LCM*)
	(natɒkborɒt)	(natɒk + borɒt)	(natɒk + borɒt)	(natɒk + borɒt)	(natɒk + borɒt)	(natɒk + bor + ɒt)

Plus (+) symbol indicates the morpheme boundaries.
EM−Emphatic marker; DM−Definitive marker; PL−Plural marker;
LCM−Locative case marker; NCM−Nominative case marker; GCM−Genitive case marker; ACM−Acquisitive case marker;

Table 3.11: Comparison of stemming by different approaches for Assamese noun root নাটক (natɒk : *drama*).

Sl.	Word	Stemmer			Morfessor	Correct stemming
		Approach 1	Approach 2	Approach 3		
(x)	কৰা	কৰা	কৰা	কৰা	কৰা	কৰা (*to do+2PPrT*)
	(kɒra)	(kɒra)	(kɒra)	(kɒra)	(kɒra)	(kɒra)
(xi)	কৰিছিলা	কৰ+ িছিলা	কৰ+ িছিলা	কৰ+ িছিলা	কৰ+ িছিলা	কৰ+ িছিলা (*to do+2PPT*)
	(kɒrisila)	(kɒr + isila)	(kɒr + isila)	(kɒr + isila)	(kɒr + isila)	(kɒr + isila)
(xii)	কৰিলাহেতেননে	কৰ+ িলাহেতেননে	কৰ+ িলাহেতেননে	কৰ+ িলাহেতেননে	কৰ+ িলাহেতেননে	কৰ+ িলা+হেতেন+নে (*to do+2PCPT+EM*)
	(kɒrilahetenne)	(kɒr + ilahetenne)	(kɒr + ilahetenne)	(kɒr + ilahetenne)	(kɒr + ilahetenne)	(kɒr + ila + heten + ne)

Plus (+) symbol indicates the morpheme boundaries.
EM–Emphatic marker;
2PPrT– 2^{nd} person, present tense; 2PPT– 2^{nd} person, past tense; 2PCPT– 2^{nd} person, conditional past tense;

Table 3.12: Comparison of stemming by different approaches for Assamese verb root কৰ (kɒr : to do).

Example: বইগুলি → বই + গুলি

IPA: bɔiguli → bɔi (*book*) + guli (*plural marker*)

(b) root + CM

Example: বইর → বই + র

IPA: bɔiɔr → bɔi (*book*) + ɔr (*genitive case marker*)

(c) root + DM

Example: বইটা → বই + টা

IPA: bɔita → bɔi (*book*) + ta (*definitive marker*)

(d) root + EM

Example: বইহে → বই + হে

IPA: bɔihɛ → bɔi (*book*) + hɛ (*emphatic marker*)

(e) root + PM + CM

Example: বইগুলির → বই + গুলি + র

IPA: bɔiguliɾ → bɔi (*book*) + guli (*plural marker*) + ɾ (*case marker*)

(f) root + PM + EM

Example: বইগুলিহে → বই + গুলি + হে

IPA: bɔigulihɛ → bɔi (*book*) + guli (*plural marker*) + hɛ (*emphatic marker*)

(g) root + CM + EM

Example: বইরহে → বই + র + হে

IPA: bɔiɾhɛ → bɔi (*book*) + ɾ (*case marker*) + hɛ (*emphatic marker*)

(h) root + DM + CM

Example: বইটার → বই + টা + র

IPA: bɔitaɾ → bɔi (*book*) + ta (definitive marker) + ɾ (*case marker*)

(i) root + DM + EM

Example: বইটা → বই + টা + হে

IPA: bɔitahɛ → bɔi (*book*) + ta (definitive marker) + hɛ (*emphatic marker*)

(j) root + PM + CM + EM

Example: বইগুলিরহে → বই + গুলি + র + হে

IPA: bɔiguliɾhɛ → bɔi (*book*) + guli (*plural marker*) + ɾ (*case marker*) + hɛ (*emphatic marker*)

(k) root + VM

Example: জানেন → জান (ো) + েন

IPA: ʤanen → ʤana (*to know*) + nen (*present tense, honorific marker*)

Using only the suffix stripping approach, we obtained 56% accuracy. We improve the accuracy to 81% on adding a frequent word list of size 30105. From corpus

analysis, we find 9 single letter ambiguous suffixes (see Appendix C) responsible for decline in the strength of the rule-based approach. We obtain nearly 10% improvement over Approach 2 on applying the hybrid approach, when tested on 1502 words.

2. **Bishnupriya Manipuri** is an Indo-Aryan language spoken in Assam, Tripura and Manipur of India as well as in Sylhet region of Bangladesh and some nearby regions of Burma, with remarkable influence from Assamese, Bengali and Meitei[17]. According to Sinha [60], in a 30000 Bishnupriya Manipuri word list, almost 4000 were of Meitei origin. Although the roots borrowed from Meitei cannot take affix directly, some Bishnupriya Manipuri root words attach after Meitei roots forming compound words; suffixes may be attached to such compound words. Sinha [60] also reported that the stable elements of the language such as declensional endings, conjugationals and pronominal forms of Bishnupriya Manipuri are of Indo-Aryan origin and are closely related to Assamese, Bengali and Oriya. Among thirty five principal phonemes, Bishnupriya Manipuri has eight vowels, twenty five consonants and two semi-vowels. The most important fact about the languages is the formation of words starting with nasal sounds like ঙ (ŋ). After manual validation of the rules generated by the rule engine, we found 8694 suffix sequences using 10 rules. The following are the rules used in rule engine.

   (a) root + PM
   Example: মানুহাবি → মানু + হাবি
   IPA: manuhabi → manu (*man*) + habi (*plural marker*)

   (b) root + CM
   Example: মানুর → মানু + র
   IPA: manur → manu (*man*) + r (*case marker*)

   (c) root + DM
   Example: মানুহান → মানু + হান
   IPA: manuhan → manu (*man*) + han (*definitive marker*)

   (d) root + EM
   Example: মানুহে → মানু + হে
   IPA: manuhɛ → manu (*man*) + hɛ (*emphatic marker*)

   (e) root + PM + CM
   Example: মানুহাবির → মানু + হাবি + র
   IPA: manuhabir → manu (*man*) + habi (*plural marker*) + r (*case marker*)

[17]Meitei or Meetei is mainly spoken in the North-east Indian state of Manipur and belongs to the Tibeto-Burman language family.

(f) root + PM + EM
Example: মানুহাবিহে → মানু + হাবি + হে
IPA: manuhabihɛ → manu (*man*) + habi (*plural marker*) + hɛ (*emphatic marker*)

(g) root + CM + EM
Example: মানুরহে → মানু + র + হে
IPA: manurhɛ → manu (*man*) + r (*case marker*) + hɛ (*emphatic marker*)

(h) root + DM + CM
Example: মানুগর → মানু + গ + র
IPA: manugɔr → manu (*man*) + gɔ (definitive marker) + r (*case marker*)

(i) root + DM + EM
Example: মানুগহে → মানু + গ + হে
IPA: manugɔhɛ → manu (*man*) + gɔ (definitive marker) + hɛ (*emphatic marker*)

(j) root + VM
Example: করইলু → কর + ইলু
IPA: kɔrɔilu → kɔr (*to do*) + ilu (*past tense marker*)

Using only the suffix stripping approach we obtained 53% accuracy. We improved the accuracy to 77%, on adding a frequent word list of size 10350. From corpus analysis, we find 10 single letter ambiguous suffixes (see Appendix C) responsible for decline in the strength of the rule-based approach. We obtain nearly 10% improvement over Approach 2 on applying the hybrid approach, when tested on 1510 words.

3. **Bodo**, a tonal language with two tones belongs to the Tibeto-Burman language family. It is spoken mainly in North-east India and has very close resemblance to Rabha, Garo, Dimasa and Kokborok[18]. Among its 22 phonemes, it has six vowels and sixteen consonant sounds. The use of the high back unrounded vowel phoneme (ɯ) is very frequent in Bodo. After manual validation of the rules generated by the rule engine we found 6344 suffix sequences using 11 rules. The following are the rules used in rule engine.

(a) root + PM
Example: बिजाबफोर → बिजाब + फोर
IPA: bizabp^{h}ɔr → bizab (*book*) + p^{h}ɔr (*plural marker*)

[18]All four languages are spoken in different places of North-east India, belong to the Tibeto-Burman language family and are vulnerable languages according to UNESCO.

(b) root + CM

Example: बिजाबनि → बिजाब + नि

IPA: bizabni → bizab (*book*) + ni (*genitive case marker*)

(c) root + DM

Example: बिजाबबो → बिजाब + बो

IPA: bizabbə → bizab (*book*) + bə (*definitive marker*)

(d) root + EM

Example: बिजाबनो → बिजाब + नो

IPA: bizabnə → bizab (*book*) + nə (*emphatic marker*)

(e) root + PM + CM

Example: बिजाबफोरनि → बिजाब + फोर + नि

IPA: bizabp^{h}ərni → bizab (*book*) + p^{h}ər (*plural marker*) +ni (*genitive case marker*)

(f) root + PM + EM

Example: बिजाबफोरनो → बिजाब + फोर + नो

IPA: bizabp^{h}ərnə → bizab (*book*) + p^{h}ər (*plural marker*) + nə (*emphatic marker*)

(g) root + CM + EM

Example: बिजाबनियानो → बिजाब + नि + नो

IPA: bizabninə → bizab (*book*) + ni (*genitive case marker*) + nə (*emphatic marker*)

(h) root + CM + DM

Example: बिजाबनिबो → बिजाब + नि + बो

IPA: bizabnibə → bizab (*book*) + ni (*genitive case marker*) + bə (*definitive marker*)

(i) root + DM + EM

Example: बिजाबबोनो → बिजाब + बो + नो

IPA: bizabbənə → bizab (*book*) + bə (*definitive marker*) + nə (*emphatic marker*)

(j) root + PM + CM + EM

Example: बिजाबफोरनियानो → बिजाब + फोर + नि + यानो

IPA: bizabp^{h}ərnijanə → bizab (*book*) + p^{h}ər (*plural marker*) + ni (*genitive case marker*) + janə (*emphatic marker*)

(k) root + VM

Example: थांबाय → थां + बाय

IPA: t^{h}aŋbai → t^{h}aŋ (*to go*) + bai (*present perfect tense marker*)

Using only the suffix stripping approach, we obtained 45% accuracy. We improved the accuracy to 71% on adding a frequent word list of size 9502. From corpus analysis, we find 10 single letter ambiguous suffixes (see Table 3.13 and Table 3.14) responsible for decline in the strength of rule-based approach. We achieve nearly 11% improvement over Approach 2 on applying the hybrid approach, when tested on 1509 words.

Suffix	Category	Inflected word	Root word
Language : Bengali			
র	Genitive CM	বৃষ্টির (bristi + r : *rain + GCM*)	বাজার (bazar : *market*)
ি/ই	Tense marker	খাই (k^{h}a + i : *eat + TM*)	বই (boi : *book*)
া/আ	Tense marker	করা (kor + a : *to do* + TM)	রাস্তা (rastta : *road*)
ে/এ	Locative CM	ওপরে (opor + e : above + LCM)	সে (ʧe *he/she*)
োা/ও	Tense marker	টুকরো (tukor + o : piece : 1PPl)	মতো (moto : *like*)
Language : Bishnupriya Manipuri			
র	Genitive CM	ফলর (p^{h}ol + or : *fruit + GCM*)	জোর (ʤor : *press*)
ছ	Tense marker	চিনছ (sin + ʧ : recognize + TM)	বাছ (baʧ : *select*)
ু/উ	Tense marker	আছু (aʧ+ u : *to be + TM*)	ইপু (ipu : *grand father*)
ি/ই	Tense marker	আছি (aʧ + i : *to be + TM*)	কলি (koli : *bud*)
া/আ	Tense marker	কাদা (kad + a : *to weep + TM*)	কলা (kola : *art*)
ে/এ	Tense marker	আছে (aʧ + e : *to be + TM*)	আমারে (amare : *us*)
Language : Bodo			
আ / া	Nominative CM	রামা (ram+a : *Ram+NCM*)	অনসুলা (ɔnsula : honest)
উ / ু	Nominal suffix		উন্দু (undu : *to sleep*)
इ /ি	Negation suffix	जायि (za+i' : *eat+do not*)	मानसि (mansi : *man*)
स	Verbal suffix	दानस (dan+sɔ : *to cut+separate*)	सिरिस (siriʧ : *name of a tree*)
ग	Verbal suffix	मावग (maw+gɔ : *work+finish*)	
ल	Emphatic marker	नोंल (nɒŋ+lɔ : *you+only*)	आखल (ak^{h}ɔl : *character*)
ख	Adjective denoting suffix	गाबख (gaw+k^{h}ɔ : *access crying*)	फिसाख (p^{h}isak^{h}ɔ : *womb*)

NCM → Nominative case marker; GCM → Genitive case marker;
LCM → Locative case marker;TM → Tense marker; 1PPl → First person plural marker.

Table 3.13: Single letter suffixes in Bengali, Bishnupriya Manipuri and Bodo with examples of inflected words and root words ending with that letter.

The obtained results for all the languages are shown in Table 3.15. The languages used in the study, except Bengali, still lack good balanced corpora. Using our rule engine we produce 12456, 8694 and 6344 suffix sequences for Bengali, Bishnupriya Manipuri and Bodo, respectively. Being verb final languages, the investigated languages have a complex morphology for the verb. The small size of the root word list may be the reason behind the

Language : Bengali	Total	%
Words with single letter suffixes (M_{sl})	864	26.90
Words with multi-letter suffixes(M_{sm})	612	19.05
Words with no suffix (N_ϵ)	1054	32.82
Root words that end with any member of the single letter suffix set (N_{sl})	682	21.23
Language : Bishnupriya Manipuri		
Words with single letter suffixes (M_{sl})	516	20.31
Words with multi-letter suffixes (M_{sm})	554	21.81
Words with no suffix (N_ϵ)	958	37.72
Root words that end with any member of the single letter suffix set (N_{sl})	512	20.16
Language : Bodo		
Words with single letter suffixes (M_{sl})	522	19.92
Words with multi-letter suffixes (M_{sm})	684	26.10
Words with no suffix (N_ϵ)	883	33.69
Root words that end with any member of the single letter suffix set (N_{sl})	532	20.29

Table 3.14: Suffix information in the training corpora.

low accuracy in Bishnupriya Manipuri and Bodo. We may be able to improve the accuracy by increasing the dictionary size and with more insights to the languages in designing the rules used by the rule engine. These two languages are vulnerable as mentioned earlier and linguistic expertise is difficult to find. For manual evaluation, we employ one evaluator for each language; the evaluators are highly educated and native speaker of the language. We compare our result with unsupervised approaches such as Dasgupta and Ng [61].

Language	SLS	A1(%)	DS	A2(%)	SSS	A3(%)
Assamese	18,194	57	25,000	81	11	94
Bengali	12,456	56	30,105	84	8	94
Bishnupriya Manipuri	8,694	53	10,350	77	10	87
Bodo	6,344	45	9,502	71	11	82

A1 → Approach 1; A2 → Approach 2; A3 → Hybrid Approach
SLS → Suffix list size; DS → Dictionary size; SSS → Single suffix size

Table 3.15: Results obtained for Assamese, Bengali, Bishnupriya Manipuri and Bodo using various approaches.

Das and Bandyopadhyay [62] and Sharma at al. [12] and the comparisons are given in Table 3.16. From the table, it is clear that our approach work well with low resource languages, particularly ones from India. We also compare our results with these obtained by Morfessor. The obtained results are shown in Table 3.16 with 123753, 130512, 42580 and 40103 words for Assamese, Bengali, Bishnupriya Manipuri and Bodo, respectively. We have to mention here that the corpora used for Bishnupriya Manipuri and Bodo are

	Assamese	Bengali	Bishnupriya	Bodo	Approach
Morfessor[58]	81%	80%	81%	78%	Unsupervised
Dasgupta & Ng[61]	-	84%	-	-	Unsupervised
Das & Bandyopadhyay[62]	-	74.06%	-	-	K-means Clustering
Sharma et al.[12]	85%	-	-	-	Unsupervised
Our hybrid Approach	94%	94%	87%	82%	Hybrid

Table 3.16: Comparison of our result with other approach

not balanced. For Bishnupriya Manipuri, texts are collected from blogs and Wikipedia, whereas for Bodo we have manually typed 40103 words for our work. In Table 3.16, the results obtained in our experiments with Morfessor and the hybrid approach are presented. Other three results shown are from the respective reports. Since the data sets (and languages) for the different approaches are not the same, small variations in the quality of the output may be ignored.

3.8 Summary

In this work, we have presented stemmers for texts in Assamese, Bengali, Bishnupriya Manipuri and Bodo. All are morphologically rich, agglutinating and relatively free word order Indian languages. First we use a rule-based approach and obtain 57%, 56%, 53% and 45% stemming accuracy, respectively. Next, we add a frequent word list to the rule-based approach and increase the accuracy substantially to 81%, 84%, 77% and 71% for the same languages, respectively. We found that for the language set, a dominant fraction of suffixes are single letter and words ending such single letters create problems during suffix stripping. Therefore, we propose a new method that combines the rule-based algorithm for predicting multiple letter suffixes and an HMM-based algorithm for predicting the single letter suffixes. The resulting algorithm uses the strengths of both algorithms leading to a much higher accuracy of 94% compared to just 82% for Assamese and 94%, 87% and 82% for Bengali, Bishnupriya Manipuri and Bodo, respectively. It is possible that named entity recognition, prior to stemming or in parallel may help. This is because many errors occur with OOV words, a lot of which are named entities. However, since languages considered (except Bengali; even Bengali researchers complain of lack of corpora and tools) are resource-poor languages, named entity recognizers are not readily available although there is some published research [63, 64].

As future work, it would be interesting to explore the possibility of modelling all morphological phenomena using other successful techniques such as Optimality Theory [65], Maximum Entropy Models [66] and Conditional Random Fields [67] and comparing the results with those of our approaches.

Chapter 4

AsmPoST: Part-of-Speech Tagger for Assamese

" ··· each part-of-speech a spark awaiting redemption, each a virtue, a power in abeyance ··· "

– The Necessity, Denise Levertov (1923 -1997)

Outline: This chapter presents the following:

1. A brief introduction to part-of-speech tagging (PoS).
2. A brief description of previous work related to PoS tagging.
3. Issues related to PoS tagging of Assamese.
4. Design of an Assamese specific PoS tagset.
5. Description of approaches used to identify the grammatical category of each word.
6. Discussion and summary.

4.1 Introduction

Part-of-speech tagging is the process of marking up words and punctuation characters in a text with appropriate PoS labels. Thus, the aim of automatic PoS tagging is to determine the lexical attributes of a given word in various contexts and situations. Two factors determine the grammatical (i.e., syntactic) category of a word. The first is lexical information directly related to the category of the word, and the other is contextual information related to the environment in which the word is used. The tagging problem can be handled either at the word level or at the sentence level. Word level tagging is a classification problem where an appropriate grammatical category is assigned to each word of a sentence, considering attributes of words that may vary from language to language. On the other hand, in sentence level tagging a series of grammatical categories are assigned one by one to the words in the sequence. The problems faced in PoS tagging are manifold.

1. Many words that occur in natural language texts are not listed in any catalogue or lexical database. We term such words as *out of vocabulary (OOV) words.*

2. A large percentage of words in a text also show *ambiguity* regarding lexical category. A single word form may relate to various mutually exclusive categories of linguistic information. For instance, the Assamese word কৰ (kɔr) is either a verb or a noun. As a noun it means *hand.* The noun form can be inflected for case, number and person. In the verb sense, it means *to do,* possess finite or non-finite forms, and the finite form can be inflected for tense, aspect and modality of the verb.

3. Insufficient training data to develop an automatic PoS tagger is a crucial problem. Most South Asian languages, unlike English and some other European languages are less computationally explored. As a result, they still lack annotated corpora or sometimes even balanced raw corpora.

Though there are a number of methods for PoS tagging, their effectiveness varies across languages. Although the main categories are language invariant, the entire set of PoS tags is generally language dependent as each language has its own distinct characteristics. To develop a PoS tagger we need one tagset, either language specific or language independent, and the tagging approach. Designing a well defined tagset is difficult. Section 4.4 describes the structure of the Assamese tagset we have designed.

Jurafsky and Martin [7] classify all PoS tagging algorithms into three categories, viz., rule based, stochastic, and hybrid. Most taggers, either rule based, stochastic or hybrid, are initially developed for English, and afterwards adapted to other languages. Brill's tagger [68], is a widely discussed linguistically motivated rule based PoS tagger for English. In the two stage architecture of Brill's tagger, input tokens are initially tagged with their most likely tags in the first stage, and lexical rules are employed to assign tags to unknown tokens in the second stage. On the other hand, TnT [69], a widely discussed statistical PoS tagger based on a second order Markov model, was developed for both English and German. It calculates the lexical probabilities of unknown words based on their suffixes. Comparison between statistical and linguistic rule based taggers shows that for the same amount of ambiguity, the error rate of a statistical tagger is an order of magnitude greater than that of the rule based one [70]. The taggers described above are specifically designed for relatively fixed word order languages, where position of the word plays an important role. Dincer et. al [71] describe a suffix based PoS tagging approach for Turkish, a relatively free word order language. Using the well-known Hidden Markov Model with a closed lexicon that consists of a fixed number of letters from the ends of words, they obtain accuracy of 90.2%.

Among Indo-Aryan languages, Sanskrit is a purely free word order language [72], but modern Indo-Aryan languages such as Hindi, Bengali and Assamese have partially lost the free word order in the course of evolution. As mentioned above, in fixed word order languages, position plays an important role in identifying the word category whereas this is not true for relatively free word order languages. Most Indian languages are morphologically rich, inflection being pre-dominant. We use Assamese as the primary target language for our experiments as an example of a modern Indian language. We also experiment with other Indian languages such as Bengali and Manipuri for comparison and to provide evidence that our approach will work similarly well for other Indian languages.

This chapter is organized as follows. In Section 4.2, we give a brief survey of literature related to PoS tagging, the state-of-the-art and linguistic characteristics of Assamese. In any language, nouns and verbs are the most crucial parts of a sentence. The noun class is always an open lexical category; its members can occur as the head word in the subject of a clause, the object of a verb, or the object of a preposition. In Section 4.5, we present our methodology and a brief description of how we identify nouns and verbs using affix information and our results (Section 4.5.4) In Section 4.6, we describe our dictionary look-up based approach to PoS tagging. The HMM is a widely used technique in the literature of PoS tagging. In Section 4.7, we discuss our experiments and results (Section 4.7.1) using HMM. Section 4.10 concludes the chapter with hints of

future work.

4.2 Related work

PoS tagging plays a pivotal role in various NLP tasks including question-answering, parsing and machine translation. It is defined as a process of assigning a label to each word in a sentence indicating the state of the word within a predefined system of syntax for that specific language.

During the last two decades, many different types of taggers have been developed, especially for European *corpus rich languages* such as English, Czech and Turkish. As discussed in Section 4.1, Brill tagger [68], TnT tagger [69] and Claw Tagger [73] are examples of some early PoS taggers for English. We have found more than 10 different techniques in the literature used in different contexts and languages. Techniques of information theory, such as maximum entropy models [74] (96.6% accuarcy for English), Hidden Markov Models [69] (96.6% accuracy for English), support vector machines [75] (97.10% for English), conditional random fields [67] (95.7% for English), artificial neural networks [76] and decision trees [77, 78] (96.36% for English) are among the popular techniques employed to train, classify and assign PoS tags.

There have been a good amount of reported work on unsupervised PoS tagging. Among these, Goldwater and Griffiths [79] describe a trigram Hidden Markov Model based approach and report 74.5% accuracy. They also use a Bayesian approach and a sparse distribution to improve performance to 83.9% and 86.6%, respectively. They conclude that the Bayesian approach is particularly helpful because learning is less constrained. Yarowsky and Ngai [80] investigate the potential for inducing multilingual PoS taggers using parallel corpora. Based on their work, an unsupervised graph-based approach is described by Das and Patrov [81] for languages that do not have labeled data, but have been translated to resource-rich languages. Their experiment with eight languages shows around 10.4% improvement over a baseline approach. Biemann [82] describes cluster-based unsupervised PoS tagging for English, Finnish and German. Unlike predefined tagsets as in the current scenario, Biemann's approach itself computes the number of tags for the entire text. They use context similarity and log-likelihood statistics as similarity measures for the clusters.

There are some techniques in the literature, which are based on the use of a dictionary. Hajic [83] develops a dictionary based morphology driven PoS tagger for five

morphologically rich languages: Romanian, Czech, Estonian, Hungarian, and Slovene and concludes that an approach based on morphological dictionaries is a better choice for inflectionally rich languages. Oflazer and Kuruoz [84] report a morphology driven rule based PoS tagger for Turkish, using a combination of hand-crafted rules and statistical learning. Shamfard and Fadaee [85] report a hybrid morphology based PoS tagger for Persian where they combine features of probabilistic and rule-based taggers to tag Persian unknown words. Ravi and Knight [86] describe a dictionary-based unsupervised approach that uses a dictionary, integer programming and entropy maximization to set parameter values. They achieve 92.3% and 96.8% accuracy with a 45-element tagset and 17-element tagset, respectively. Umansky-Pesin [87] describes a web based model to enhance the quality of PoS tagging for unknown words. They integrate their approach with MXPOST [74] and experiment with English, Chinese and German. They reduce tagging error by 16.63%, 13.57% and 18.09%, respectively over the output of the original MXPOST tagger. Georgiev et al. [88] describe a guided learning framework [89] for Bulgarian, a morphologically rich Slavic language. Using 680 morpho-syntactic tags, they obtain 97.98% accuracy for Bulgarian text. Habash ad Rambow [90] show that embedding a morphological analyser to a PoS tagger outperforms the state-of-the-art for Arabic.

Due to relative free word order, agglutinative nature, lack of resources and the general lateness in entering the computational linguistics field, reported tagger development work on Indian languages is relatively scanty. Morphological richness and relatively free word order nature of Indian languages make morphological analysis a crucial task in tagging of Indian language texts. While in some Indic languages such as Assamese, most case markers occur as suffixes, in others such as Hindi they occur as separate words, leading to local word grouping. Beyond that Indian languages have similar degrees of free word orderness.

Among published work, Dandapat et. al. [104] develop a hybrid model of PoS tagging by combining both supervised and unsupervised stochastic techniques. Avinesh and Karthik [96] use Conditional Random Fields (CRF) and Transformation based Learning (TBL). The heart of the system developed by Singh et al. [92] for Hindi is the detailed linguistic analysis of morpho-syntactic phenomena. Saha et al. [105] develop a system for machine assisted PoS tagging of Bangla corpora. Pammi and Prahllad [106] develop a PoS tagger and chunker using decision forests. This work explores different methods for PoS tagging of Indian languages using sub-words as units. Singh and Bandyopadhyay [97] develop a morphology driven PoS tagger for Manipuri with accuracy of 65% for single tagged correct words. Table 4.1 shows some reported PoS taggers in Indian languages.

Report	Year	Technique	Tagset	Accuracy	Language
Ray et al. [72]	2003	Lexical sequence constraint, constraint propagation, morphological and ontological information	-	-	Hindi
Dandapat et al. [91]	2004	Supervised and unsupervised learning using HMM and morphological analyser	-	95.00%	Bengali
Singh et al. [92]	2006	CN2 algorithm	-	93.45%	Hindi
Dalal et al. [93]	2006	Maximum Entropy Markov model	29 Tags	88.40%	Hindi
Hellwig [94]	2007	HMM and *Sandhi* rules	136 Tags	-	Sanskrit
Sastry et al. [95]	2007	HMM based tagging using TnT	26 Tags	78.35%	Hindi
				74.58%	Bengali
				75.27%	Telugu
PVS & G. [96]	2007	CRF for initial tagging and transformation based rules to correct errors produced by CRF	26 Tags	78.66%	Hindi
				76.08%	Bengali
				77.37%	Telugu
Singh & Bandyopadhyay [97]	2008	Morphology driven PoS tagger using root word, prefix and suffix dictionary	26 Tags	69.00%	Manipuri
Shrivastava & Bhattacharyya [98]	2008	HMM with naive stemming for preprocessing	-	93.12%	Hindi
Manju et al. [99]	2009	HMM and rule based approach	18 Tags	90%	Malayalam
Sharma & Lehal [100]	2011	HMM based model	630 Tags	90.11%	Punjabi
Gupta et al. [101]	2011	Rule based TENGRAM method (based on rough set theory)	8 Tags	-	Hindi
Reddy & Sharoff [102]	2012	Cross lingual PoS tagging of Kannada using Telugu as base system	-	-	Kannada
Ekbal & Saha [103]	2012	Single and multi-objective optimization conditional random field and support vector machine (simulated annealing based ensemble technique)	26 tags	87.67%-89.88%	Hindi, Bengali

Table 4.1: Some reported PoS tagging approaches in Indian languages

4.3 Linguistic issues

Though Assamese is relatively free word order, the predominant word order is SOV (subject-object-verb). In Assamese, secondary forms of words are formed through affixation (inflection and derivation), and compounding [12]. Affixes play a very important role in word formation. Affixes are used in the formation of relational nouns and pronouns, and in the inflection of verbs with respect to number, person, tense, aspect and mood. For example, Table 4.2 shows how a relational noun দেউতা (deuta : father) is inflected depending on number and person.

Person	**Singular**	**Plural**
প্রথম পুৰুষ	মোৰ দেউতা	আমাৰ দেউতা
(prɔt^{h}ɔm purus : 1^{st} person)	(mor deuta : My father)	(amar deuta : Our father)
মান্য মধ্যম পুৰুষ	তোমাৰ দেউতাৱা	তোমালোকৰ দেউতাৱা
(manjɔ mad^{h}jɔm purus : 2^{nd} person, Honorific)	(tomar deutara : Your father)	(tomalokɒr deutara : Your father)
তুচ্ছ মধ্যম পুৰুষ	তোৰ দেউতাৰ	তহঁতৰ দেউতাৰ
(tutʃɔ mad^{h}jɔm purus : 2^{nd} person, Familiar)	(tor deutar : Your father)	(tɒh$\tilde{}$tɒr deutar : Your father)
তৃতীয় পুৰুষ	তাৰ দেউতাক	সিহঁতৰ দেউতাক
(tritij purus : 3^{rd} person)	(tar deutak : Her father)	(sih$\tilde{}$tɒr deutak : Their father)

Table 4.2: Personal definitives are inflected for person and number

There are 5 tenses in Assamese [107], namely, present, past, future, present perfect and past perfect tense. Besides these, every root verb changes with case, tense, person, mood and honorificity. The following paragraphs describe just a few of many characteristics of Assamese text that make the tagging task complex.

1. Suffixation of nouns is very extensive in Assamese. There are more than 100 suffixes for the Assamese noun. These are mostly placed singly, but sometimes in sequence after the root word.

2. We need special care for honorific particles like ডাঙৰীয়া (daŋrija : *Mr.*). Assamese and other Indian languages have a practice of adding particles such as দেউ (deu : *Mr.*), মহোদয় (mɒhodɒj : *sir*), মহোদয়া (mɒhodɒja : *madam*), মহাশয় (mɒhasɒj : *sir*), and মহাশয়া (mɒhasɒja : *madam*), after proper nouns or personal pronouns. They are added to indicate respect to the person being addressed.

3. Use of foreign words is also common in Assamese. Often regular suffixes of Assamese are used with such words. Such foreign words will be tagged as per the syntactic function of the word in the sentence.

4. An affix denoting number, gender or person, can be added to an adjective or other category word to create a noun word. For example-
   ধুনীয়াজনী হৈ আহিছা।
   IPA : dʰunijazɒni hoi ahisa
   Here ধুনীয়া (dʰunija : *beautiful*) is an adjective, but after adding feminine definitive জনী (zɒni) the whole constituent becomes a noun word. Table 4.3 shows some other examples of formation of derived words in Assamese.

Prefix	Word	Category	Suffix	New word	Category
-	আঙুলি (aŋuli-*finger*)	NN	-আ (a)	আঙুলিয়া (aŋulia-*process of pointing*)	VB
-	কৰ (kɒr *to do*)	VB	-আ (a)	কৰা (kɒra *do, 2nd person*)	VB
-	কৰা (kɒra *do, 2nd person*)	VB	-জন (zɒn)	কৰাজন (kɒrazɒn *the worker*)	NN
ন (nɒ)	কৰা (kɒra *do, 2nd person*)	VB	-জন zɒn)	নকৰাজন (nɒkɒrazɒn)	NN
-	চল (sɒl *to move*)	VB	-অন (ɒn)	চলন (sɒlɒn *process of moving*)	NN
-	ডাঙৰ (daŋɒr *elder*)	NOM	-জনী (zɒni)	ডাঙৰজনী (daŋɒrzɒni *the elder one (female)*)	NN
অপ (ɒp)	শক্তি (sɒkti *energy*)	NN	-	অপশক্তি (ɒpsɒkti)	NN

Table 4.3: Formation of derivational noun and verb in Assamese

5. Even conjunctions can be used as other parts of speech.
   হৰি আৰু যদু ভায়েক-ককায়েক।
   IPA : hɒri aru jɒdu bʰajek kɒkajek.
   ET : Hari and Jadu are brothers.

   কালিৰ ঘটনাটোৱে বিষয়টোক আৰু অধিক ৰহস্যজনক কৰি তুলিলে।
   IPA : kalir gʰɒtɒnatowe bisɒjtok aru adʰik rɒhɒisjɒzɒnɒk kɒri tulile.
   ET : Yesterday's incident has made the matter even more mysterious.

   The word আৰু (aru : *and*) shows ambiguity in these two sentences. In the first, it is used as conjunction and in the second, it is used as an adverb.

6. Depending on the context, even a common word may have different PoS tags. For example: If কাৰণে (karɒne), দৰে (dɒre), নিমিত্তে (nimitte) and হেতু (hetu) are placed after a pronominal adjective, they are considered conjunction and if placed after noun or personal pronoun they are considered particle. For example,
   এই কাৰণে মই নগ'লো।
   IPA: ei karɒne mɒi nɒgɒlo.
   ET: This is why I did not go.

   ৰামৰ কাৰণে মই নগ'লো।
   IPA: ramɒr karɒne mɒi nɒgɒlo.
   ET : I did not go because of Ram.
   In the first sentence কাৰণে (karɒne) is placed after pronominal adjective এই (ei); so karɒne is considered conjunction. But in the second sentence কাৰণে (karɒne) is placed after noun ৰাম (*ram*), and hence karɒne is considered particle.

The above examples reflect the ambiguity in word and context level tagging. Complexities in Examples 1, 3 and 4 are based on affixation. Ambiguities in Examples 5 and 6 are based on context, where the same word has different meanings based on different usages. The next section will describe state-of-the-art Assamese tagsets and our work on tagset development.

4.4 Assamese part-of-speech tagset

The first requirement of any PoS tagger is the tagset or a well-defined list of grammatical categories to help in disambiguating word senses by humans as well as by machines. There

are a number of tagsets such as the *PENN*[1] tagset (Appendix B.1), the *BNC*[2] tagset (Appendix B.2) and the *Claw*[3] tagset, developed based on intended applications, with particular views and requirements in mind. Each has its pros and cons. For example, the above mentioned tagsets were designed based on different views for English. One decade after the development of the *PENN tagset* (1993), a common tagset for Indian languages was developed at *IIIT Hyderbad* (2006). It contains 26 tags. The tagsets that we discuss in this section are of two kinds – those which follow global standards and guidelines and those which evolve from particular language specific requirements. Tagsets such as ILPoST, ILMT and LDC-IL are designed to take care of all Indian languages, and this could provide resource sharing and reusability of linguistic resources. Flat tagsets may be easier to process but they cannot capture higher level granularity without an extremely large list of independent tags. They are hard to adapt too. In languages like Assamese, morphological and other grammatical attributes are step-wise/level-wise/hierarchy-wise appended one after another to the root. We can say that structure-wise the inflections are hierarchical. Thus as obvious, to annotate a hierarchical structure hierarchical tagset covers all the attributes of a morphologically rich agglutinative language.

The following section discusses the tagset development in the Indian language context related to Assamese.

4.4.1 TUTagset-F

During 2007-08, a flat (F) tagset[4] was developed at Tezpur University solely for Assamese. This is a large tagset with 172 tags. It has separate tags for general case markers as well as very specific noun case markers. For example, the tag *NCM* is used for nominative case marker and *CN1* and *CNS1* are used for nominative singular common noun and nominative plural common noun, respectively. Appendix B.4 shows details of this tagset called TUTagset-F.

[1]http://www.mozart-oz.org/mogul/doc/lager/brill-tagger/penn.html
[2]http://ucrel.lancs.ac.uk/bnc2/bnc2guide.htm#tagset
[3]http://ucrel.lancs.ac.uk/claws1tags.html
[4]http://tezu.ernet.in/~nlp/posf.pdf

4.4.2 Xobdo tagset

Simultaneous to the TUTagset-F, Xobdo, an Assamese online dictionary project, had developed another Assamese flat tagset[5]. This project uses the same tagset for English, Hindi, Bengali and all north-east Indian languages including Bodo, Karbi and Khasi. This tagset contains 15 tags. It groups all case endings, prefixes and suffixes as adpositions. Though Assamese has a rich system of particles, Xobdo excludes particles other than the ones for interjection and conjunction. Appendix B.3 provides the details of the Xobdo tagset.

4.4.3 LDC-IL tagset

Linguistic Data Consortium for Indian Languages (LDC-IL) is a consortium for the development of language technology in India and was set up by a collective effort of the Central Institute of Indian Languages, Mysore, and several other like-minded institutions working on Indian language technologies. LDC-IL tagset is considered as a standard tagset for annotating the corpora of Indian languages. It has a total of 26 tags. Appendix B.5 provides the LDC-IL tagset.

4.4.4 Description of our tagset : *TUTagset-H*

In designing a hierarchical (H) tagset, we follow the desgning guidelines for *AnnCora* (Bharati et al. 2006) [108], Penn tagset and *MSRI-JNU Sanskrit tagset*[6]. We use the same tags as in the Penn Treebank when possible, so that they are easily understandable to all annotators. The tags designed in the course of our owrk for Assamese are shown in Table 4.4.

Sl.	Symbol	1^{st} level	2^{nd} level	Example
1.	NN	Noun	Common	মানুহ (manuh : *man*), তৰা (tɒra : *star*), কিতাপ (kitap : *book*)
			Proper	তেজপুৰ (tɛzpur : *Tezpur*), অসম (ɔxɒm : *Assam*)
			Material	চকী (sɒki : *chair*), মেজ (mɛz : *table*)

[5]http://xobdo.org/dic/help/pos.php
[6]http://sanskrit.jnu.ac.in/corpora/MSR-JNU-Sanskrit-Guidelines.htm

		Abstract	বিষাদ (bixad : *sadness*), আনন্দ (anɒndɒ : *happiness*)
		Verbal	ভ্রমণ (bʰrɒmɒn : *travel*), মৰণ (mɒrɒn : *death*)
		Time Indicative	পুৱা (puwa : *morning*), গধুলি (gɒdʰuli : *evening*)
		Group Indicative	সভা (sɒbʰa : *meeting*), সমিতি (sɒmiti : *committee*)
2. PN	Pronoun	Personal	সি (si : *he*), তাই (tai : *she*)
		Reflexive	নিজ (niz : *own*), স্বয়ং (sɒjɔŋ : *self*)
		Reciprocal	যি (যিজনক) (zi : *who*), যিহ্ (যাক) (zak : *whom*)
		Inclusive	সদৌ (sɒdɔu : *all*), সকলো (sɒkɒlo : *all*)
3. VB	Verb	Infinite	কৰ (kɒr : *to do*), যা (za : *to go*)
		Finite	কৰিলোঁ (kɒrilo :), চালোঁ (:)
		Causative	কৰোৱা (kɒrowa :), ধুওৱা (:)
4. RB	Adverb	Time	আজি (aji : *today*), কালি (kali : *yesterday*)
		Location	তাত (tat : *there*), ইয়াত (ijat : *here*)
		Manner	
5. NOM	Nominal Modifier	Adjective	চলন্ত (sɒlɒntɒ : *moving*), খৰকৈ (kʰɒrɒkoi : *quickly*)
		Demonstrative	এইটো (ɛito :), সেইটো (sɛito : *that*)
		Pre-nominal	শ্রী (sri : *sri*), ডঃ (dɔktɒr : *doctor*)
6. PAR	Particle	Conjunction	আৰু (aru : *and*), কাৰণে (karɒnɛ : *because of*)
		Disjunction	নাইবা (naiba : *or*), তথাপি (tɒtʰapi : *even*)
		Exclamatory	যদি (zɒdi : *if*), কিজানি (kizani : *possibly*)
		Vocative	ঐ (oi : *ditto*), হেৰি (hɛri : *hey*)
		Particle	দৰে (dɒrɛ : *way*), মতে (mɒtɛ : *accordingly*)
7. PSP	Post-position	Post-position	দ্বাৰা (dara : *by*), সৈতে (sɒitɛ : *with*)
8. QH	Question word	Interrogative Pronoun	কি (ki : *what*), কোন (kon : *who*)

		Interrogative Particle	কেনেকুৱা (kenekuwa : *how*), নেকি (neki :)
9. RDP	Reduplication	Reduplicative	তামোল-চামোল (tamol-samol : *betel nut etc.*), মাছ-তাছ (mas-tas : *fish etc.*)
		Onomatopoeic	অকাই-পকাই (ɔkai-pɒkai : *convolve*), পালী-পহৰী (pali-pɒhɒri : *watchman*)
		Echo Word	কা-কা (ka-ka : *voice of crow*)
10.NUM	Number	Cardinal	
		Ordinal	
		Date	১-বহাগ-১৯১৪ (:), ২৯-৭-৮২ (:)
		Time	১-১৫ ঘণ্টা (:), পুৱা ৭:১৫ (:)
11.EXT	Extra	Symbols	%, $, @, +
		Abbreviation	অগপ (:), আছু (:)
		Unit of Measure	কি.মি. (:), কিলো (:)
		Foreign word	
12.PUN	Punctuation		।, !, ?, ;
13.UNK	Unknown word		

Table 4.4: Assamese hierarchical tagset

The following is the explanation of first level tags

1. **NN: Noun**
   All nouns including proper nouns fall into this category. A Noun may take markers for gender, number, case, emphasis, definiteness, inclusiveness, exclusiveness, topic, confirmation, and honorificity. For example,

   (a) মানুহজনী আহিছিল।
   মানুহজনী (manuhzɒni : *the lady*)_NN আহিছিল (ahisil : *went*)_VB ।_PUN.

2. **PN: Pronoun**
   The Penn tagset does not have one single tag for pronouns. However, we have decided to use a pronoun tag to cover all pronouns in our coarse-grained tagset. This is because our coarse-grained tagset mostly corresponds to the standard part-of-speech. For example মই (mɒi : *me*), তই (tɒi : *you*), সিহঁত (sihɒt : *they*), এইবোৰ (eibɒr : *these*), and সেইবোৰ (seibɒr : *those*).

3. **VB: Verb**
   We label verbs of all kinds as VB at the coarse-grained level. Due to the inflectional

property of Assamese, assigning tags to verbs is a complex job. Verbs are modified with tense, aspect and mood. Assamese uses three verbs to serve the purposes of verb 'be' in English. These are আছ (asɒ : *to be at some position or time*), থাক (tʰak : *to stay*), and হ (hɒ : *to be something or someone*). No form of verb 'be' is used as copula in affirmative sentences in the present tense. However, it is present in future tense, past tense and in negative constructions in present tense. Table 4.5 shows the use of the 'be' verb. The suffixation of verbs is to some extent regular. There are four irregular verbs, নাই (*nai* : *not to be*), আছ (asɒ : *to have*), থাক (tʰak : *to stay*), and ব'ল (bɔl : *to go*), whose inflections are not bounded with regular rules.

Present Tense	Present tense – Negative	Past Tense	Future Tense
তেওঁ মোৰ স্ত্রী	তেওঁ মোৰ স্ত্রী নহয়	তেওঁ মোৰ স্ত্রী আছিল	তেওঁ মোৰ স্ত্রী হব
teo͂ mor stri.	teo͂ mor stri nɒhɒj	teo͂ mor stri asil	teo͂ mor stri hɒbɒ
(she) (my) (wife)	(she) (my) (wife) (is not)	(she) (my) (wife) (was)	(she) (my) (wife) (will be)
She is my wife.	She is not my wife.	She was my wife.	She will be my wife.

Table 4.5: Examples of use of হ (hɒ : *to be*) verb in Assamese.

4. **RB: Adverb**

From the positional point of view, the Assamese adverb is always placed before the verb. However, sometimes the object of the verb may be placed between verb and adverb. According to definition, an adverb is *a word that modifies something other than a noun.* Some adverbs commonly used in Assamese are ততালিকে (tɒtalike : *instantly*), আকৌ (akou : *again*), সদায় (sɒdaj : *always*), নিতৌ (nitɒu : *daily*), তুৰন্তে (turɒnte : *immediately*), কেলেই (kelei : *why*), নিচেই (nisei : *very*), দুনাই (dunai : *again*), অকলশৰে (ɒkɒlsɒre : *alone*), কাচিৎ (kasit : *occasionally*), তৎক্ষণাৎ (tɒtkʰjɒnat : *immediately*), হঠাৎ (hɒtʰat : *suddenly*), প্ৰায়ে (praje : *regularly*), কদাচিত (kɒdasit : *never*), অথনি (ɒtʰɒni : *a little earlier*), বেগাই (begai : *speedily*), সোনকালে (sonkale : *quickly*), ঘনাই (gʰɒnai : *frequently*), and পুনৰ (punɒr : *again*). Some adverbs in Assamese are formed, after adding some specific post-positions with numerals, pronouns, adjectives and reduplicative words. Here are some examples.

(b) ডাঙৰকৈ নাহাঁহিবা।
ডাঙৰকৈ(daŋɒrkɒi : *loudly*)_RB নাহাঁহিবা(nahahiba : *do not laugh*)_VB I_PUN.

(c) ভূমিকম্পত হাজাৰে হাজাৰে মানুহ মৰিল।
ভূমিকম্পত(bʰumikɒmpɒt : *in earthquake*)_NN হাজাৰে(hazare : *thousand*)_RB হাজাৰে(hazare : *thousand*)_NOM মানুহ(manuh : *man*) মৰিল(mɒril : *died*)_VB I_PUN.

(d) অথনিকৈ আহিবা।
অথনিকৈ(ɒtʰɒnikɒi : *a little later*)_RB আহিবা(ahiba : *come*)_VB I_PUN.

(e) আৰু কোৱা।
আৰু(aru : *more*)_RB কোৱা(kowa : *tell*)_VB I_PUN.

In Example 4b and Example 4d, ডাঙৰকৈ (daŋɒrkɒi : *loudly*) and অথনিকৈ (ɒtʰɒnikɒi : *a little later*) are two adverbs placed before the verb নাহাঁহিবা (nahahiba : *do not laugh*) and

আহিবা (ahiba : *come*). In Example 4e, the conjunction particle আৰু (more : *and*) is used as adverb.

5. **NOM: Nominal modifier**

Traditional adjectives, demonstratives and pre-nominals are nothing but modifiers of nouns or noun phrase. In the Penn tagset, there are a number of tags for nominal modifiers of a sentence, including adjectives (JJ), comparative adjective (JJR), superlative adjective (JJS), and a preposition (IN). In *AnnCora* also, there are a number of tags for nominal modifiers such as demonstrative (DEM), adjective (JJ), classifiers (CL) and intensifier (INTF). In the coarse-grained level, we decide to keep the entire information under one tag Nominal modifier (NOM). Here are some examples.

(f) ধুনীয়া ফুল (beautiful flower)
ধুনীয়া (d^{h}unia : *beautiful*)_NOM ফুল(p^{h}ul : *flower*)_NN

(g) বৰ ধুনীয়া ফুল (very beautiful flower)
বৰ(bɒr : *very*)_RB ধুনীয়া (d^{h}unia : *beautiful*)_NOM ফুল(p^{h}ul : *flower*)_NN

(h) সৌ লৰাটো (that boy)
সৌ(sɒu : *that*)_NOM লৰাটো(lɒrato : *the boy*)_NN

6. **PSP: Post-positions**

All Indian languages have the phenomenon of postpositions. A post-position expresses certain grammatical functions such as case, plurality, and definiteness. Examples includes দ্বাৰা (dara : *by*), সৈতে (sɒite : *with*) and পৰা (pɒra : *from*).

7. **RP: Particle**

Assamese has a large number of particles. Depending on semantics, an Assamese particle can be classified into eight different groups. Particles never change form. Here are some examples.

(i) মই আৰু তুমি যাম।
মই (mɒi : *I*)_PN আৰু (aru : and)_RP তুমি (tumi : *you*)_PN যাম(zam : *will go*)_VB ।_PUN.

(j) ছিঃ ! তেনে কথা কব নাপায়।
ছিঃ(sih :) _RP!_PUN তেনে(tene : *like*)_RP কথা(kɒt^{h}a :)_NN কব(kɒbɒ : *speak*)_VB নাপায়(napaj : *do not*)_VB ।_PUN

(k) তুমি মাতা হেতুকে মই আহিলোঁ।
তুমি(tumi : *you*)_PN মাতা(mata : *call*)_VB হেতুকে(hetuke :)_RB মই(mɒi : *I*)_PN আহিলোঁ(ahilo : *came*)_VB ।_PUN

8. **QW: Question word**

Penn Treebank has four tags for question word, (WDT, WP, WP_x, WRB), and the BNC

tagset has also three question tags (DTQ, AVQ, PNQ). As all question words in Assamese start with the character ক (k), we can term these words *k*–words of Assamese, following the term *wh*–word in English. For example - কি (ki : *what*), কিহে (kihe : *what*), কোনে (kone : *who*), কিয় (kio : *why*), কাৰ (kar : *whose*), কত (kɔt : *where*), কেতিয়া (ketija : *when*) and কেনেকৈ (kenekɒi : *how*).

Question words are inflected with case markers and plural markers or a sequence of both. Sometime particles like বোলে (bole), হয় (hɒj), জানো (zano), and হেনো (heno) are used as question words. For example-

(l) সি বোলে গ'লগৈ ?
সি(si : *he*)_PN বোলে(bole : *is said*)_QW গ'লগৈ(gɒlgɒi :)_VB ?_PUN

(m) তই(tɒi : *you*)_PN পিকনিকলৈ(piknikɒlɒi : *to picnic*)_NN যাবি(zabi : *will go*)_VB জানো(zano :)_QW ?_PUN.

(n) তুমি(tumi : *you*)_above PN গৈছিলা(gɒichila : *went*)_VB নেকি(neki :)_QW ?_PUN.

The same set of words mentioned above may express doubt and may belong to a particle category, depending on the tone used. There are situations where suffix like নে (ne) is added after the main verb to form question. Again, the tone becomes important here. Consider the following situations.

(o) বোপা ঘৰত আছানে ?
বোপা(bopa : *brother*)_ ঘৰত($g^hɒrɒt$: *at home*) আছানে(asane :)_QW(VB+QW) ?_PUN

(p) কোন কোন আহিছাহক ?
কোন(kon : *who*)_QW কোন(kon : *who*)_QW আহিছাহক(ahisahɒk :)_VB ?_PUN

The second example above is interesting, where the question word (For example কোন (kon : *who*)) is repeated to mark the plural sense.

9. **NUM: Number**

All numbers including date, time and units belong to this category. Problems arise when we get a date or time with embedded text characters. For example, ১লা বহাগ (1la bɒhag : *first day of Assamese month Bohag*)

10. **RDP: Reduplication**

This is a special phenomenon in most Indian languages, where either the same word is written twice to indicate emphasis (for example- কা-কা (ka-ka), কৰ-কৰ (kɐr-kɐr), হায়-হায় (hai-hai)) or using a rhyming nonsense word after a regular lexical word indicating the sense of '*etc.*' (for example- মাছ-তাছ (mas-tas), কিতাপ-চিতাপ (kitap-sitap)) or using a rhyming actual word after another regular word (for example- অকাই-পকাই (ɒkai-pɒkai), পালি-পহৰি (pali-pɒhɒri)).

11. **EXT: Extra**
    This is a tag for foreign words; special symbols like #, $ $, &, % and @; acronyms like অগপ (AGP) and বিজেপি (BJP); and an important category, units of measures like কি.মি. (K.M.) and কিলমিটাৰ (Kilometer).

12. **PUN: Punctuation**
    All punctuations such as Devanagari danda (।, ॥) called দাড়ি (dari) in Assamese, question mark (?) and brackets ((,},[) belong to this category.

13. **UNK: Unknown word**
    Most Indian languages have a number of loan words from other languages. A loan word written in its own script or in Roman script belongs to this category. Also in cases such as when the annotator is not sure of the category of a word, it can also be tagged as UNK.

This tagset covers *single word tokens, named entities of various types, compound word tokens, abbreviations* and *punctuations*. The first two types include items that belong to common dictionaries, followed by items that refer to real world entities and the last type *punctuation* contains text formatting items. The design of our annotation scheme does not rely only on linguistic assumptions of traditional Assamese grammar, but also on the output needed for further linguistic processing of data. The next section discusses a suffix based noun and verb categorization approach for Assamese.

4.5 Suffix based noun and verb categorization[7]

As Assamese nouns and verbs are open lexical categories, if we can tag words in these classes correctly, the problem of tagging the remaining words in a text will be alleviated. A preliminary survey by Sharma et al. [12] showed that about 48% of words in an Assamese text of around 1,600 words were inflectional or derivational whereas only about 19% words in an English text of about 1,400 words were so. Similarly, in a sample Hindi text of about 1,000 words, 26% were inflectional and derivational. To make it easy, we categorise Assamese words into an inflected class and a non-inflected class. In the inflected class noun inflections and verb inflections are predominant. The next two subsections give a detailed glimpse of noun and verb morphology.

[7]This section of the thesis is published in Proceedings of the International Conference on Asian Language Processing, pp:19–22, Harbin, China, 2010, IEEE

4.5.1 Noun morphology

Noun inflection represents gender, number and case in Assamese. For example, nouns ending with -জন (zɒn) and -জনী (zɒni) are identified as masculine and feminine nouns, respectively. All rules applied to noun inflection can be applied to pronouns, adjectives and even numerals (with a few exceptions). Table 4.6 and Table 4.3 show formation of compound and derivational words, respectively. Most compound words in Assamese are nouns; although other forms are also not rare. Derivations take place using either suffixes or prefixes, or a combinations of both. The base for derivation can be a simple word or a compound word. It is observed that only suffixes can change the word category, prefixes do not change the category of a word.

Stem(Category)	Stem(Category)	New word(Category)
কথা (NN)	ছবি (NN)	কথাছবি (NN)
(kɒt^{h}a : *talk*)	(sɒbi : *picture*)	(kɒt^{h}asɒbi : *cinema*)
কৃষ্ণ (NN)	পক্ষ (NN)	কৃষ্ণপক্ষ (NN)
(krisnɒ : *dark*)	(pɒk^{h}jɒ : *fortnight*)	(krisnɒpɒk^{h}jɒ : *dark fortnight*)
অন্ধ (NN)	বিশ্বাস (NN)	অন্ধবিশ্বাস (NN)
(ɒnd^{h}ɒ : *blind*)	(biswas : *belief*)	(ɒnd^{h}ɒbiswas : *superstition*)

Table 4.6: Formation of compound words in Assamese

4.5.2 Verb morphology

Assamese verbs are inflected with tense, aspect, modality and honorificity. Traditionally, Assamese verbs are categorised as either *finite* or *non-finite*. Verb roots are in non-finite form to which tense, person or grammatical markers are added. In comparison to nouns, the Assamese verb inflection is complex. Sharma et al. [11, 12] reported 520 inflectional forms for root verb বহ্ (bɒh : *to sit*).

4.5.3 Morphology driven PoS tagging

Knowing how a suffix is used, for example, to a word W, when suffix X is attached to either nominal or verbal root R, it is possible to identify the category of R based on the category of X (e.g., whether it is a member of the nominal group or verbal group). We use local linguistic knowledge to filter out the wrong category, e.g., whether W is a proper noun or common noun or whether the noun category is animate or inanimate or whether the verb category is finite or not. For example, any word that ends with -পাৰা (-para), -বাৰী (-bari), -গাঁও (gão), চুবুৰী (suburi) and -গড় (-gɒr) are names of places, as these are among most popular suffixes placed

after village or neighbourhood names. In our method, we follow the following three basic steps to tag tokenized text.

1. **Brute-force determination of suffix sequences:** *In this step we obtain all possible sequences of noun suffixes following our method described in Section 3.4.* Assamese nouns and pronouns can take more than one suffix in a sequence, though not all orderings of the same set of suffixes are grammatically correct. An example of a suffix sequence is
   নাতিনীয়েককেইজনীমানেহে→নাতিনী + য়েক + কেইজনী + মান + ে + হে
   natinijekkɛidʒɒnimanɛhɛ→natini + jɛk + kɛidʒɒni + man + ɛ + hɛ
   nAtinIyekkeijanImAnehe → granddaughter + inflected form of kinship noun+ indefinite feminine marker + plural marker + nominative case marker + emphatic marker.

   Some suffixes are always used with words from a small class of roots. For example, the suffix -জোপা (zopa) is always placed after a word denoting a tree-like structure. The rule-engine (see Section 3.4) generates all valid and invalid sequences. So in the next step, i.e., in sequence pruning we try to minimize the search space.

2. **Suffix sequence pruning:** *In this step we filter out the invalid suffix sequences from the sequence list obtained in Step 1.* Though a number of suffixes can theoretically occur after a root word, we find, on average three suffixes and at most five suffixes append sequentially after a root in the corpus we studied. All suffix sequences are not valid. So if we list most valid suffix sequences a priori using our linguistic knowledge, we need not go through all possible combinations of the suffixes. A Java module is employed to prune the suffix sequence.

3. **Suffix stripping:** *In this step, we identify the noun and verb roots based on the single suffix that occurs immediately after the root.* For example, if we find the word মানুহবোৰৰ (manuhborɒr), it will first identify the suffix sequence বোৰৰ (borɒr) (বোৰ+ৰ: plural marker + genitive marker), which is a noun suffix and hence মানুহবোৰৰ (manuhborɒr) is tagged as *genitive plural noun* and মানুহ as noun, in suffix-free form. We find that the suffix 'এ' (nominative case marker for noun and the endings of 2^{nd} person (familiar) past and present perfect tense marker) is the most ambiguous as it applies to both nouns and verbs.

4.5.4 Results and discussion

Table 4.7 shows the results obtained. We are using standard notations for computations of precision and recall. Recall counts is the number of words correctly tagged divided by total number of words, whereas precision is the ratio of correctly tagged words among tagged words. For example, out of 100 noun words in the test data, if the tagger tagged 50 words as noun words and out of the fifty words tagged as noun only 30 words are actually noun. That is,

the tagger incorrectly tags 20 words as nouns, whereas they are from other category. Thus precision of the tagger will be 0.60 (30/50) and recall will be 0.30 (30/100). Most of the time, foreign words are written in a transliterated form, and depending on the word category in the source language, suffixation occurs. It is also observed that most transliterated words originate

Category	No. of words in test data	Precision	Recall	F-measure
Noun	28923	0.87	0.79	0.82
Verb	2629	0.91	0.87	0.88

Table 4.7: Precision, recall and F-measure of Approach-1

in English and are of noun category. In such a situation, we obtain the best results using the proposed approach, whereas most of the approaches discussed in literature survey, fail to handle unknown words such as transliterated words, words written using other scripts (such as Roman, Bengali, etc.) with Assamese suffix such as UGCৰ. We found two situations, where the proposed approach may fail.

- A review of around 2,000 words using *single character suffixes*, shows that we have only three pairs of single letter inflections that share same character for two inflectional categories. For example ক (kɒ) is a symbol of acquisitive marker as well as 3^{rd} person present tense marker. In such a case, the proposed algorithm fails to identify the appropriate category. (see Table 3.5)
- There are some words like প্ৰয়োজন (prɒjozɒn : *need*), কিছুমান (kisuman : *some*), and কেতবোৰ (ketbor : *some*) that end with some nominal suffix, though they are neither inflected nor derived. Looking at the suffixes -জন (-zɒn : *definiteness marker*), -মান (-man : *indefiniteness marker*), -বোৰ (bor : *plural marker*) the proposed algorithm will identify the words as noun, though they are not.

To resolve the above drawbacks we can use a list of frequent words, in the form of a dictionary. The next section discusses a dictionary based approach to improve the accuracy and to extend the scope of tagging to other categories.

4.6 Incorporating dictionary to enhance accuracy[8]

As mentioned above, Assamese verbs and nouns are open class word categories whereas other categories such as pronouns, particles, nominal modifiers (adjectives and demonstratives) and

[8]This section of the thesis is published as a book chapter in Machine Intelligence: Recent Advances, Narosa Publishing House, Editors. B. Nath, U. Sharma and D.K. Bhattacharyya, ISBN-978-81-8487-140-1, 2011

adverbs are small and closed. In Assamese, most words less than 4 characters long have more than one meaning [11]. Our dictionary contains root words and their corresponding tags and prefix and suffix information, which show ambiguity at word level. We reuse the rule engine described in Chapter 3 (Section 3.4), that generates all possible suffix sequences for Assamese. A Java module obtains all possible suffix sequences for a dictionary word and tags them. The dictionary is used primarily to reduce the amount of ambiguity. Table 4.8 describes the dictionary entries. Our dictionary file contains only 10,456 entries with tags.

Category	Pronoun	Verb	Noun	Particle	Adverb	NOM	Post-position	**Total**
Size	89	881	4955	162	392	3967	10	**10456**

Table 4.8: Word list information used in dictionary-based PoS tagging. NOM : Nominal modifier

Algorithm 1 Algorithm for dictionary-based PoS tagging.

Input: A dictionary file, a suffix file and a corpus *crps*
Output: Tagged Corpus

1: Read dictionary and suffix file and store them in separate arrays
2: *for* Each token in the corpus *crps do*
3: *if* the token is in dictionary file,
4: Tag it with corresponding tag against the dictionary element.
5: *elseif* The token ends with any element of suffix file,
6: Tag the token with corresponding tag against the suffix.
7: *else* The token starts with any element of prefix file,
8: Tag the token with corresponding tag against the prefix.
9: Check whether tagged token satisfies the hand-crafted rules or not.

We use Java to implement this algorithm. The results obtained are shown in Table 4.9. To resolve ambiguities such as noun-adjective ambiguity, noun-verb ambiguity, and adjective-adverb ambiguity, we use a small rule base. The rules we use are listed below.

1. Adverbs always precede verbs and adjectives precede nouns.
2. Words ending with -কৈ (-koi) are generally adverbs.
3. Words ending with plural markers or definitives are always noun.
4. Except single constituent sentences, particles do not occur in the initial position of a sentence.

Our approach is based on affix information regarding words and categories of root words. We resolve ambiguity at the context level also. Suppose we get more than one tag for a specific token. In such a case, we check the previous token t_1 and using a hand-crafted rule we mark

the token t_2 and check the next token t_3. We backtrack to t_2 to determine whether it is correct considering the tag on t_3. To some extent, this simple procedure covers contextual information also. However, a problem will arise if t_3 also has more than one tag.

4.6.1 Results and discussion

The results of our tagging experiment are given in Table 4.9. In a text containing 38,314 words, the tagger correctly tags 35,647 words. We find 8.14% error rate from this experiment with 10,456 dictionary words. That is the accuracy is about 91.86%. Information about morphological features and contextual features is used to resolve the OOV words. Morphological features such as affix information and other context rules determine the tag for the word. Let us consider the output text in Figure 4.1. Here four words, viz., গৱর্ণমেন্ট (*government*), মেট্রিক (*metric*), ইন্টাৰমেডিয়েট (*intermediate*) and ডিগ্রী (*degree*) are marked as OOV. All four words are English words written in Assamese script without a morphological marker. Therefore the algorithm cannot detect the category and marks them as unknown words. In the same figure, the word কলেজৰ (kɒlɛʤɒr) is also an English word written in Assamese script and marked as noun because the English word কলেজ (*college*) is associated with the genitive case marker ৰ. A manual verification of the tagged text has been carried out after the tagging process is over. This is the first step towards developing a rule based tagging approach for Assamese, and we

তেজপুৰ<NN> গৱর্ণমেন্ট<OOV> হাইস্কুলৰ<NN> পৰা<PAR>
১৯৪০<NUM> চনত<NN> প্ৰৱেশিকা<OOV> ,<PUN>
১৯৪২<NUM> চনত<NN> কটন<NN> কলেজৰ<NN>
পৰা<PAR> ইণ্টাৰমেডিয়েট<OOV> ,<PUN> ১৯৪৪<NUM>
চনত<NN> বেনাৰচ<NN> হিন্দু<NN> কলেজৰ<NN>
পৰা<PAR> স্নাতক<NN> আৰু<PAR> ১৯৪৫<NUM>
চনত<NN> ৰাজনীতি<NN> বিজ্ঞানত<NN> স্নাতকোত্তৰ<NN>
ডিগ্রী<OOV> লাভ<Noun> কৰে<VB> ।<PUN>

Figure 4.1: Example output

believe modifying some of our rules or creating additional rules will increase the performance of the tagger. We compare our result with published results in other languages in Table 4.10. A 24,000 entry lexicon is used to tag Turkish text [84], whereas Singh and Bandyopadhyay [97] use only 2,051 entries in the dictionary to tag Manipuri text. Hajic et al. [83] report 3.72% error rate for Czech, 8.20% for Estonian, 5.64% for Hungarian, 5.04% for Romanian and 5.12% for Slovene. As mentioned above, we use a dictionary of size 10,456 words and obtain around 91% accuracy. Thus our work is comparable to other morphology driven approaches.

PoS label	Actual size	Correctly tagged	Error rate
NN	22176	18459	16.76
PN	1517	1398	07.84
NOM	4176	4012	03.93
RB	440	428	02.73
VB	1622	1319	18.68
NUM	170	164	03.53
PAR	2817	2698	04.22
PUN	5333	5333	00.00
PSP	9	9	00.00
Other	54	50	07.41

Table 4.9: Label-wise error rate obtained by incorporating dictionary

Author	Language	Dictionary size	Accuracy
Oflazer and Kuruoz [84]	Turkish	24,000	98.00%
Singh and Bandyopadhyay [97]	Manipuri	2,051	69.00%
Hajic [83]	Romanian		94.96%
	Czech		96.28%
	Hungarian		94.36%
	Estonia		91.80%
	Slovene		94.88%

Table 4.10: Results compared with other dictionary based work.

4.7 HMM based PoS tagging[9]

The intuition behind all stochastic taggers is a simple generalization of the "*Pick the most likely tag for this word*" approach [7]. For a given word sequence, HMM taggers choose the best tag, and then the best tag sequence that maximizes the following formula:

$$P(word|tag) * P(tag|previous\ n\ tags).$$

In the HMM model, an "emission probability" is the probability of observing the input sentence or sequence of words W given the state sequence T, that is $P(W|T)$. Also, from the state transition probabilities we can calculate the probability $P(T)$ of forming the state sequence T. The "transition probability" from one state to another state is defined by $P(n_i|n_{i-1}n_{i-2})$. Formally, for a sequence of words W the problem is to find the tag sequence T, which maximizes

[9]This section of the thesis is published in Proceedings of the ACL-IJCNLP Conference Short Paper, pp:33-36, Singapore, 2009, ACL

the probability $P(T|W)$ is,

$$argmax P(T|W).$$

Using Bayes' rule for conditional probabilities, we can calculate P(T|W) as

$$P(T|W) = [P(W|T) * P(T)]/P(W).$$

Since the probability of the word sequence $P(W)$ is the same for each sequence, we can ignore it giving $P(N|W) = P(W|N) * P(N)$ and, hence, $P(N|W)$ needs to be maximized as

$$argmax P(W|T) * P(T).$$

We use HMM [109] and the Viterbi algorithm [57] in developing our PoS tagger. Irrespective of word order, whether it is SOV, SOV or other, based the on evidence available in corpus/training set, the HMM model assigned tags to words. If the frequency of SOV ordered sentences are more, obviously it will assign tags like the SOV structure not like VOS or OSV. First, in the training phase, we manually tag parts of the EMILLE, Assamese Pratidin and Wikipedia corpora using the flat and hierarchical TUTagsets discussed above. Then, we build four database tables using probabilities extracted from the manually tagged corpus-word-probability table, previous-tag-probability table, starting-tag-probability table and affix-probability table. For testing, we consider three text segments, A–part of the EMILLE corpus, B–part of the Assamese Pratidin corpus, and C–part of the Wikipedia corpus, each of about 3500 words. First the input text is segmented into sentences. Each sentence is parsed individually. Each word of a sentence is stored in an array. After that, each word is looked up in the word-probability table. If the word is unknown, its possible affixes are extracted and looked up in the affix-probability table. From this search, we obtain the probable tags and their corresponding probabilities for each word. All these probable tags and the corresponding probabilities are stored in a two dimensional array, which we call the *lattice* of the sentence. If we do not get probable tags and probabilities for a certain word from these two tables, we assign the tag of the common noun (CN in flat tagset) and probability 1 to the word since CNs occur most frequently in the manually tagged corpus. After forming the lattice, the Viterbi algorithm is applied to the lattice to yield the most probable tag sequence for the sentence. After that the next sentence is taken and the same procedure is repeated.

4.7.1 Results and discussion

We manually tag Assamese text from three different corpora. The Assamese Pratidin corpus is a news corpus whereas the Wikipedia corpus is a collection of Wikipedia articles. The training corpus consists of nearly 10,000 words. We use around five thousand words (taken from the three

corpora) to test the tagger. Due to the morphological richness of the language, many words of Assamese occur in secondary forms in texts. This increases the number of PoS tags needed for the language. Each main category in flat tagset is sub-categorised based on six case endings (viz, nominative, accusative, instrumental, dative, genitive and locative) and two numbers. The results using the three test segments are summarised in Table 4.15. The evaluation of the results requires intensive manual verification effort. More reliable results can be obtained using larger test corpora. The first test set consist of 1,697 words from the EMILLE corpus. Out of these, 564 (including 192 proper nouns) are unknown words. The tagger correctly tagged 339 (including 68 proper nouns) unknown words. Likewise, the second and third test sets consist of 1,881 and 1,432 words, including 590 and 494 unknown words from the Assamese Pratidin and the Wikipedia corpora, respectively.

Set	Size	Using flat tagset			Using hierarchical tagset		
		Accuracy	UWH	UPH	Accuracy	UWH	UPH
A	1697	81.55%	60.10%	35.41%	85.68%	62.81%	40.10%
B	1881	86.12%	42.45%	53.96%	89.94%	59.54%	53.36%
C	1432	88.05%	54.63%	36.52%	90.45%	61.64%	46.47%

Size of training words : 10,000;
UWH : Unknown word handling accuracy;
UPH : Unknown proper noun handling accuracy.
A : EMILLE corpus; B : Assamese Pratidin corpus; C : Wikipedia corpus;

Table 4.11: PoS tagging results with flat and hierarchical tagset.

Author	Language	Average accuracy
Toutanova et al. [110]	English	97.24%
Banko and Moore [111]	English	96.55%
Dandapat and Sarkar [112]	Bengali	84.37%
Rao et al. [113]	Hindi	79.50%
	Bengali	74.40%
	Telegu	58.20%
Rao and Yarowsky [114]	Hindi	76.68%
	Bengali	74.20%
	Telegu	76.01%
Sastry et al. [95]	Hindi	78.35%
	Bengali	74.58%
	Telegu	75.27%
Ekbal et al. [115]	Hindi	82.05%
	Bengali	90.90%
	Telegu	63.93%

Table 4.12: Comparison of our PoS tagging result with other HMM based models.

Table 4.12 compares our result with other HMM based reported work. From the table it is clear that Toutanova et al. [110] obtain the best result for English, 97.24%. We obtain 81-88% accuracy using the flat tagset and 85-90% accuracy using the hierarchical tagset, which is comparable to other reported works. The accuracy of a tagger depends on the size of tagset used, the vocabulary used, and the size, genre and the quality of the corpus used. Table 4.15 shows that the average accuracy obtained using the hierarchical tagset is better than the results obtained using the flat tagset. Our flat tagset containing 172 tags is rather big compared to other Indian language tagsets, whereas the hierarchical tagset consists of only 13 tags for first level. A smaller tagset is likely to give more accurate result, but may give less information about word structure and ambiguity.

4.8 Experiments in other languages

In previous sections of this chapter, we discussed three different techniques of PoS tagging for Assamese. The first technique that is suffix based noun and verb tagging approach identify nouns and verbs with 0.82 (precision 0.87) and 0.88 (precision 0.91) f-measure accordingly. Now-a-days, one very common word formation process is transliteration, where imported words change only the script and can take the suffixes of the native language. We identified some ambiguous suffixes/suffix sequences, which may occur in nouns as well as in verbs. Therefore, to reduce ambiguity we use a handmade dictionary as a second approach. As the third task, we experiment with HMM to tag Assamese text and obtain 0.88 precision in average of hierarchical and 0.85 precision for flat tasget. Looking at the robustness and on obtaining excellent results in Assamese, we extend our approach to three other languages from Eastern India. The two other factors that behoove us to experiment with other north-eastern languages are-

1. To test, whether the proposed approaches will equally work well for other languages.
2. There are a little reported work in the literature for the resource-poor languages of north-east India. We have not come across any such computational work related to PoS tagging in the literature for these languages. Therefore we consider Bishnupriya Manipuri and Bodo language for experimen, which are listed as vulnerable languages by UNESCO.

Our experiments attempt to furnish the above factors in the context of Bengali, Bishnupriya Manipuri and Bodo languages in Chapter 3.

Approach I: Exploiting the inflectional information for Identification of open categories – Considered languages for experiments are highly inflectional and agglutinative in nature. Being agglutinative languages, the suffixes may append one after another sequentially.

Digging deeper the grammar of languages, we find that, prefix does not encode the information of word category, rather suffix plays an important role in discovering the word category. Again, if we maintain a dictionary with assigned category to each word may mislead the tagger to assign a wrong tag in case of **category migration** after suffixing (For example, see Example no. 4 of Section 4.3, Page no. 50).

	AS			BN			BPY			BD		
	S	P	R	S	P	R	S	P	R	S	P	R
Noun	28923	0.87	0.79	28167	0.88	0.79	15510	0.81	0.74	12140	0.79	0.84
Verb	02629	0.91	0.87	01956	0.92	0.85	01040	0.82	0.83	00820	0.83	0.81

S=Number of words, P=Precision, R=Recall
AS=Assamese, BN=Bengali, BPY=Bishnupriya Manipuri, BD=Bodo

Table 4.13: PoS tagging results for Assamese, Bengali, Bishnupriya Manipuri and Bodo using suffix based noun verb identification approach

The obtained results for all the languages using suffix-based noun verb identification are shown in Table 4.13. We use the knowledge of suffix information discussed in the earlier chapter. Identification of these two categories based on suffix information may reduce the computational or processing cost as these two categories are open lexical category. The robustness of this approach is that, for any new inflected word to the language the approach can easily identify the category of the word.

Approach II: Incorporating dictionary – As mentioned earlier, verbs and nouns are open class word categories for the considered languages whereas other categories such as pronouns, particles, nominal modifiers (adjectives and demonstratives) and adverbs are small and closed. We exploit this property and use a list of lexicon for each language, which we term dictionary. The size of dictionary used in experiments for Assamese, Bengali, Bishnupriya Manipuri and Bodo are 10456, 12114, 7676 and 7186 accordingly. This approach correctly tagged 28528 out of 32972, 27119 out of 31963, 24002 out of 27284, and 22702 out of 26067 words (excluding punctuation and post-position words) for Assamese, Bengali, Bishnupriya Manipuri and Bodo accordingly. Label-wise obtained error-rates after incorporating dictionary are shown in Table 4.14. It is observed (Table 4.14) that, in each situation the error rate of identifying the noun and verb are higher than any other category in the list. The two reasons of high error rate may be-

- Word-based tagging do not check the contextual information. This situation arises when one word have more than one tag, which implies more than one meaning. For example, the word কৰ (kɒr) have two PoS tags with more than three meanings.

  Word – – Category – Meaning
  কৰ (kɒr) –– Noun –– Hand
  কৰ (kɒr) –– Noun –– Tax

কৰ (kɒr) –– Noun –– Bud
কৰ (kɒr) –– Noun –– Bud
কৰ (kɒr) –– Verb –– To do

- There are situations where same suffix is used to denote different category. For example, suffix -ক (-kɒ) is used as acquisitive case marker that denotes noun and 2^{nd} person present tense marker that denotes verb. Thus it raises confusion regarding the actual category of the word.

  ৰামক (ramɒk) = ৰাম (ram) + ক + (ɒk) = ram + acquisitive case marker = noun

  কৰক (kɒrɒk) = কৰ (kɒr) + ক + (ɒk) = kɒr + present tense marker = verb

	AS			BN			BPY			BD		
L	S	C	E	S	C	E	S	C	E	S	C	E
NN	22176	18459	16.76	23245	18810	19.08	20616	17710	14.10	20088	17054	15.10
PN	01517	01398	07.84	01285	01192	07.24	00812	00751	07.51	00512	00481	06.05
NOM	04176	04012	03.93	04212	04086	02.99	03104	02983	03.90	03028	02862	05.48
RB	00440	00428	02.73	00310	00297	04.19	00280	00266	05.00	00266	00246	07.52
VB	01622	01319	18.68	01240	01082	12.74	01078	00917	14.94	00890	00804	09.66
NUM	00170	00164	03.53	00121	00118	02.48	00084	00079	05.95	00078	00072	07.69
PAR	02817	02698	04.22	01525	01510	00.98	01280	01268	00.94	01170	01152	01.54
Other	00054	00050	07.41	00025	00024	04.00	00030	00028	06.67	00035	00031	11.43

L=PoS label, S=Number of words, C=Correctly tagged word, E=Error rate [((S–C)/S)*100)]
AS=Assamese, BN=Bengali, BPY=Bishnupriya Manipuri, BD=Bodo

Table 4.14: Label-wise obtained error-rates after incorporating dictionary for Assamese, Bengali, Bishnupriya Manipuri and Bodo language

Approach III: Treat context with HMM – For a given word sequence, HMM taggers choose the best tag, and then the best tag sequence that maximizes the following formula.

$$P(word|tag) * P(tag|previous\ n\ tags).$$

We use this approach with Viterbi algorithm to handle the contextual information. The obtained results by using HMM with flat and hierarchical tagset for the considered languages are shown in Table 4.15. We prepare the training dataset by manually assigning the tags to each word of sentences. For evaluation, we employ one evaluator for each language; the evaluators are highly educated and native speakers of the languages.

	TR	TS	Flat tagset			Hierarchical tagset		
			AC	UWH	UPH	AC	UWH	UPH
AS	10000	1881	86.12	42.45	53.96	89.94	59.54	53.36
BN	10000	1560	85.78	40.19	50.76	90.02	45.48	56.83
BPY	06900	1538	82.66	43.32	47.34	85.41	48.91	54.12
BD	06700	1592	79.67	37.33	48.58	84.86	45.83	51.02

TR=Training data size, TS=Test data size, AC=Overall accuracy,
UWH=Unknown word handling accuracy,
UPH=Unknown proper noun handling accuracy,
AS=Assamese, BN=Bengali, BPY=Bishnupriya Manipuri, BD=Bodo.

Table 4.15: Results obtained by using HMM with flat and hierarchical tagset for Assamese, Bengali, Bishnupriya Manipuri and Bodo language.

4.9 Tagging Multi-word unit

The term Multi-word unit (MWU) implies that it is an expression consisting of more than one word, a unit where word level and group level (i.e., contextual) information comes closer and sometimes produces a meaning different from the meaning of each of the individual expression members. Sag et. al. [116] define MWU as *any word combination for which the syntactic or semantic properties of the whole expression cannot be obtained from its parts.* For example, in English phrasal verbs (such as getting rid of, instead of), and compound nouns (such as World War II, traffic signal) are multi-word units. Thus it is the immediate next level after processing word level information. The problems discussed in Section 4.2 like tagging of reduplicated words and tagging of compound words falls into this category. So failing to identify MWU units create problems for further syntactic and semantic processing. We observed three ways of writing these units in the corpora.

Type I: As a single word. For example: Greenhouse, Policeman.

Type II: As a hyphenated word. For example: Green-house, Dining-hall.

Type III: As separate words. For example: Green house, traffic signal.

From a multi-word point of view, Type I and Type II styles of writing do not cause difficulty in processing text. They are only one word long and are supposed to be handled properly and tagged accordingly. Problems arise in Type III writing style. Currently, MWUs are receiving focus of NLP researcher due to the lowered accuracy of information extraction, parsing and question answering systems. For example, suppose an user is interested in the details of "*green house*" on a text search engine. If both words are not considered a unit or if there is no module to handle these as a single unit, the search engine may produce documents related to either

"*green*" or "*house*" only. Likewise, we may get unexpected results for units such as *driving license, traffic signal* and *bombay biking.* In this section we cover identification and processing of Assamese MWUs. For reduplication, we develop an interface to extract reduplicative MWUs from annotated and unannotated corpora. To identify and extract compound and complex units, we employ the Support Vector Machines [117] based classification tool Yamcha [2] to learn to classify the units.

Identification and extraction of multi-word units from a text corpus was one of the main areas of focus during the last decade among researchers. A number of approaches have been devised based on problems they faced, differing in terms of the type of MWUs such as compounds [118], conjuncts or complex predicates [119, 120, 121] and collocations [122], the language to which they apply such as English [122, 119, 123], Chinese [124], Turkish [125], Dutch [126] and Japanese [127], and the sources of information they use. Work reported in [128, 129, 130] focused only on independent MWUs, irrespective of type and language. Attia et al. [131] categorised the approaches into four broad categories.

- Statistical methods that use association measures to rank MWU candidates [132];
- Rule based methods that use morpho-syntactic patterns [133, 129];
- Hybrid methods which use both statistical measures and linguistic filters [134, 130]
- Word alignment methods [135, 136, 137].

Agarwal et al. [138] devise an approach for extraction of multi-word units from a Bengali untagged corpus. They use a morphological analyzer and a lexicon for pre-processing. Based on the co-occurrence frequency and a ranking function, they determine multi-word units. Dandapat et al. [139] describe an approach for identification and extraction of noun-verb collocations as multi-word units from an untagged corpus. They use a PoS tagger, a morphological analyser and compute association scores to determine the multi-word units. A compound noun MWU identification system that uses log-likelihood-ratio to rank collocations for Hindi was developed by Kunchukuttan and Damani [140]. They also use hyphenation and compound-forms of words for rank collocation. Among 80,000 words they identified 350 compound nouns as multi-word units. Their method gives 80% recall and 23% precision at rank 1000. Venkatapathy et al. [141] discuss noun-verb collocation extraction problem using MaxEnt classifier. Mukerjee et al. [142] used PoS projection from English to Hindi with corpus alignment for extracting complex predicates. Chakrabarti et al. [143] present a method for extracting Hindi compound verbs (V+V) using linguistic knowledge. A step-wise extraction of Hindi multi-word units is discussed by Sinha [144] from the machine translation viewpoint. Chakraborti et al. [145] present an approach to identifying and extracting bigram noun-noun multi-word units from a Bengali corpus by clustering semantically related nouns. They also use a vector space model for similarity

measurement. Baldwin et al. [146] reported 8% falling in accuracy in parsing of the BNC corpus using a random sample of 20,000 sentences due to miss-processing of MWUs. For our experiments, we categorise all MWU's into three groups.

1. Reduplication: For example বাট-ঘাট (bat-g^{h}at : *every nook and corner*).
2. Compound noun: For example কৃষ্ণপক্ষ (krisnɔpɔk^{h}ɔjɔ : *dark fortnight*), কাঠৰ-পুতলা (kat^{h}ɒr putɒla : *one who is entirely led by others*).
3. Compound and conjunct verbs: For example দেও উঠা (deo ut^{h}a : *to act in an unnatural way*)

4.9.1 Reduplications

- Complete reduplicatives: Here a single word or clause is repeated to form a single unit. For example লাহে লাহে (lahe lahe), and বাৰে বাৰে (bare bare).
- Mimic reduplicatives: These are naturally created sounds as perceived by speakers of a specific language. These are a kind of complete reduplication with onomatopoeic words. For example, কা কা (ka ka), and কৰ কৰ (kɒr kɒr)
- Echo reduplicatives: Sometimes called nonsense words, where the second word is simply an echo form, formed on the basis of pronunciation of the first word. For example মাছ তাছ (mas tas), and কল চল (kɒl sɒl)
- Partial reduplicatives: In partial reduplication, the second word carries some part of the first word as an affix to the second word, either as a suffix or a prefix. For example ডা ডাঙৰীয়া (da daŋɾia), and পালি পহৰীয়া (pali pɒhɒɾia) পা পইচা (pa pɒisa)

Identification of complete and mimic reduplicatives, in raw text is easy as the first word repeats for the second time. Using regular expressions, we extract all reduplicatives with no error. Echo and partial reduplication are complex processes in comparison to the first two. In echo reduplicatives, either the first few characters or the last few characters are repeated in the second word and the number of characters in both the words are the same. Therefore, we check whether (`second word length / 2`) the first half or the second half of a word match the following word. If matched, we group the word pair as a single unit. Also, another criterion is that if the first word starts with a consonant except 'চ'/ 'ছ', the next word starts with 'চ'/ 'ছ' and if the first word starts with 'চ' / 'ছ', the next word starts with either 'ত' or 'প'. The number of occurrences of partial reduplication is a bit lower, compared to the previous three units. Figure 4.2 explains the architecture of rule based reduplication identifier for an unannotated corpus. In our tagset, we have a tag called RDP, for replicated words. When we find

	Annotated				Unannotated			
	Complete	Mimic	Echo	Partial	Complete	Mimic	Echo	Partial
P	1.00	1.00	1.00	1.00	1.00	1.00	0.94	0.87

Table 4.16: Result of automatic extraction of reduplication in annotated and unannotated text

two replicated words, we tag them as RDP. For example consider the tagged text for the sentence

লাহে লাহে যাবা।
লাহে(lahe : *slowly*)_RDP লাহে(lahe: *slowly*)_RDP যাবা(zaba : *to go + future tense marker*)_VB

During grouping, when we find two RDP tags together we mark them as a single reduplicative unit. This RDP tag of our tagset, minimizes our MWU processing cost in the annotated corpus. in Table 4.16 we summarise the results of our experiments over 5000 sentences.

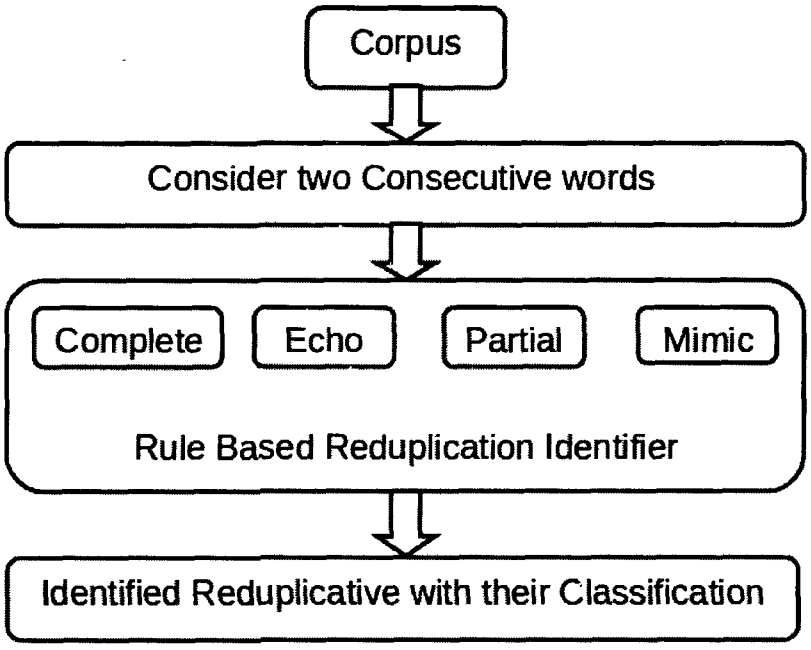


Figure 4.2: Architecture of rule based reduplication identifier for unannotated corpus.

4.9.2 Compound nouns

A compound noun consists of more than one word, that includes a noun modified by one or more modifiers. This nominal modifier includes adjectives and demonstratives or sometimes a noun

itself. Compound nouns often have a meaning that is different from the constituent words. We can define a compound noun as a sequence of more than one nouns that are connected through an implicit modification relation and work together as a complete functional unit syntactically as well as semantically [147, 148]. This definition can be divided into two parts. The first part says that it has any sequence of nouns, the second part restricts the sequence that it should work as a single functional unit. The definition excludes the noun-noun sequence. Let us consider a news headline. হীৰেন ভট্টাচার্য্যলৈ অসম উপত্যকা বঁটা (hirɛn bʰɒʧasarʤjɒloi ɒsɒm upɒtjɒka bõta). This news headline consists of five noun words, none the them form compounds, but there is a relation between the first two words and the last three words. We term these as Named Entity (NE), which we can define as a noun-noun sequence, where the *left most noun functions as a name of the rightmost noun recursively.* The first two words in the example denote the name of a famous Assamese poet with accusative case marker and the last three words (interestingly none of the words are inflected) is the name of a award given to the poet. This observation raises two questions.

As we are in a transition period, from PoS tagging to parsing, how important is it to handle these sequences of nouns as well as verbs in parsing? Although it is an identification problem, identifying noun or verb group is quite important for further processing.

At present we have three separate problems to handle: identification of compounds, identification of named entity units and identification of serialization. The question is how important is it to categorise the nominal group to a fine grained level during parsing?

Interestingly, inflectional elements such as case markers and plural markers are consistently applied to the head of the compound. The head represents the compound as a whole, and the inflections do not affect to the head alone. This is evident if we compare headless compounds with a noun or verb having an irregular form. For example, the plural of tooth is teeth, which is an irregular form retained from old English, but the plural of bluetooth is bluetooths, and not blueteeth. Compound nouns also show a stress pattern, which is distinct from other noun phrases, the stress being left-prominent, at least in English and Hindi. All these indicate that compound nouns are syntactic words. Thus, they satisfy a necessary condition for being MWUs. Digging the literature, we find nine different rules for the construction of compound nouns:

- noun+noun; Example: কাঠৰ পুতলা (kat^{h}ɒr putɒla : *one who is entirely led by others*)
- noun+verb; Example: কপাল ফুল (kɒpal p^{h}ul : *the coming of good luck*)
- verb+noun; Example: উঠি ৰজা (ut^{h}i rɒza : *a man of power and influence*)
- adjective+noun; Example: গেলা গপ (gela gɒp : *vain boasting*)
- adjective+verb; Example: তিতা দিয়া (tita dija : *to disgust greatly*)

- adverb+noun; Example: ওপৰ হাত (opɒr hat : *powerful*)
- adverb+verb; Example: দেও উঠা (deo ut^{h}a : *to act in an unnatural way*)
- noun+adjective; Example: হাত টান (hat tan : *close fisted*) and
- numeral+noun; Example: বাৰ কুৰি (barɒ kurɪ : *numerous*).

Sometimes compound nouns that consist of more than two words can be constructed recursively by combining two words at a time. Combining "science" and "fiction", and then combining the resulting compound with "writer", one can construct the compound "science fiction writer".

4.9.3 Compound and conjunct verbs

When a sentence has two or more verbs in sequence, we say that the sentence may have a compound verb. They normally consist of a main verb preceded or followed by one or more helper verbs. This type of units can be called a conjunct verb, and consists of a noun or an adjective followed by verb or vice versa. The verb bears the tense, aspect and mood information in the form of inflection. Complex and compound verbs as multi-word units have been studied widely [149, 150, 142, 151, 143]. A complex predicate is a multi-word unit where a noun, a verb or an adjective is followed by a helper or light verb and the MWU behaves as a single verb unit. A helper verb [150] can also be a main verb. A compound verb form has the main verb in its root/stem form followed by conjugated light verbs. In Assamese compound verbs, the primary meaning of the helper verbs is often completely lost and may lead to a new semantic interpretation or result affecting tense, aspect and modality of the compound verb.

4.9.4 Results and discussion

We use the Inside Outside Beginning (IOB) [1, 152] framework to identify and classify MWUs in the annotated corpus. In general, contextual information is often used as the basic feature type; the other features can then be derived based on the surrounding words, such as words and their POS tags. The chunk tag of the current token is mainly determined by the context information. For chunk tagging, we employ Yamcha [2], a supervised support vector machine based approach using polynomial kernels of degree two. We label each sentence with standard IOB tags. Since this is a binary classification task, we have 5 different tags.

- B-L : Beginning of a chunk

- I-L : Inside of a chunk
- B-I : Beginning an Idiomatic chunk
- I-I : Inside an Idiomatic chunk
- O : Outside of a chunk.

For example, let us consider an Assamese sentence.

এই	NOM	O
বছৰ	NN	O
হীৰেন	NN	B-I
ভট্টাচার্য্যলৈ	NN	I-I
অসম	NN	B-I
উপত্যকা	NN	I-I
বঁটা	NN	I-I
আগবঢ়োৱা	Adv	O
হয়	VB	O

The SVM is a binary classifier, therefore researchers have extended SVMs to perform multi-class classification. There is a popular method to extend a binary classification task to perform classification into multiple classes. It involves performing one against all classification for each of the classes. For the training feature set, the SVM uses the information available in the surrounding context, such as the specific words, their part-of-speech tags as well as chunk labels. More precisely, we use the following features to identify the group label g_i for the i^{th} word:

Word	w_{i-2}	w_{i-1}	w_i	w_{i+1}	w_{i+2}
POS	p_{i-2}	p_{i-1}	p_i	p_{i+1}	p_{i+2}
Group	g_{i-2}	g_{i-1}	g_i	...	...

w_i is the word appearing at i^{th} position, p_i is the PoS tag of the word, and g_i is the group label for i^{th} word. Since the preceding group labels for forward parsing are not given in the test data, they are obtained dynamically during the tagging of group labels. The technique can be regarded as a sort of dynamic matching, where the best answer is found by maximizing the total certainty score for the combination of tags.

	Compound noun	Compound verb	Conjunct verb
Precision	81.12	72.54	74.22
Recall	70.82	69.38	69.50

4.10 Summary

We have achieved good PoS tagging results for Assamese, a fairly widely spoken language, which has very little prior computational linguistic work. First, we implement a suffix based noun and verb tagging approach for Assamese. We obtain around 94% accuracy in identifying nouns and verbs when testing with texts containing more than 5,000 inflected noun and 5000 inflected verbs. Except noun and verb, other grammatical categories are closed class and relatively small like most languages. As the second task, we extend the noun and verb identification approach to PoS tagging using an ambiguous and frequent word list. We obtain around 91% accuracy for the dictionary based approach with a word list of10456 words. After increasing the dictionary size to 30000, we achieve 95% precision for the same. We implement an HMM based tagger as the third task and get around 87% precision with approximately 10000 training words. As the first work on Assamese, the accuracy looks good, in comparison with work in other languages. We also studied the different issues related to MWUs. We achieve 95.82%, 81.12%, 72.54% and 74.22% accuracy for reduplications, compound nouns, compound verbs and conjunct verbs respectively. From our experiments, it is clear that good features can significantly improve system performance. Our error analysis also shows that many errors are inherently ambiguous when explained using local features .

Our main achievement is the development of taggers with 87%-95% precision and creation of the Assamese tagsets. We implement an existing method for PoS tagging, but our work is for a new language where an annotated corpus and a pre-defined tagset were not available. As the word order of Assamese is relatively free, we cannot use positional information like in fixed word order languages. Another important observation from this experiment is that though Assamese is relatively free word order, some parts of speech do not occur in the initial or final positions in a sentence. We believe that embedding linguistic word agreement rules in tagging will improve the performance of the method.

Chapter 5

Parsing Assamese Text

" Grammar, which knows how to control even kings ··· "

– Les Femmes Savantes, Molière (1622 - 1673)

Outline: This chapter presents the following.

1. A brief introduction to parsing.
2. The state of the art of the relatively free word order language parsing.
3. Description of some dependency parsing approaches.
4. Description of techniques used to parse Assamese text.
5. A comparative analysis of the output.
6. Discussion and summary of the chapter.

5.1 Introduction

Parsing is a problem in many natural language processing tasks such as machine translation, information extraction and question answering. It is the term used to describe the process of automatically building syntactic analysis of a sentence in terms of a given grammar and lexicon; and syntax is the name given to the study of the form, positioning, and grouping of the elements that make up sentences. The result may be used as input to a process of semantic interpretation. The output of parsing is something logically equivalent to a tree, displaying dominance and precedence relations between constituents of a sentence. The study of natural language grammar dates back at least to 400 BC, when Panini described Sanskrit grammar, but the formal computational study of grammar can be said to start in the 1950s with work on context free grammar (CFG). Now-a-days there are several dimensions to characterize the behaviour of parsing techniques, depending on search strategy (for example - top-down or bottom-up parsing), statistical model used (for example - Maximum Entropy model) and Grammar formalism used (for example - Paninian framework). Among them most successful linguistically motivated formalisms are Combinatory Categorial Grammar (CCG) [153], Dependency Grammar (DG) [154], Lexical Functional Grammar (LFG) [155], Tree-Adjoining Grammar (TAG) [156], Head-Driven Phrase Structure Grammar (HPSG) [157], Paninian Grammar (PG) [108] and Maximum Entropy model (EM) [158]. Like some other Indo-Iranian languages (a branch of Indo-European language group) such as Hindi, Bengali (from Indic group), Tamil (from Dadric group), Assamese is a morphologically rich, relatively free word order language. Despite possessing all characteristics of a free word order language, Assamese has some other characteristics which make parsing a more difficult job.

It is worth mentioning that the majority of the parsing techniques are developed solely for the English language. As reported by Covington [6], English has almost no variation in order of constituents and it is considered a fixed word order language. Covington reports a word order variation table (see Table 5.1) and indicates that Warlpiri has the maximum variation in word order. The languages that have maximum variation can be termed free word order languages. As there is no measure to calculate the degree of variation, we use the term "relative" to denote free or fixed word orderness. In fixed word order languages, the position gives us some important information regarding the structure of the sentences. For example, while scanning an English sentence, if we find "a/an/the" the next word must be a noun word. This type of prediction reduces the computational complexity in each step of processing. It is not the case in free or relatively free word order languages. Unlike fixed word order language such as English, in morphologically rich free word order languages the preferred linguistics rule set is very large, which may be difficult to handle using approaches like PSG and LFG **Among reported formalisms, only CCG, PG and DG have been successfully applied to free word order languages** (see Table 5.2).

Almost no variation	English, Chinese, French
Some variation	Japanese, German, Finnish
Extensive variation	Russian, Korean, Latin
Maximum variation	Warlpiri

Table 5.1: Word order variation table [6].

5.2 Dependency Grammar Formalism

Much recent recent work on parsing [159, 160, 161, 162] has focus on dependency parsing. A dependency relation is an asymmetrical relation between a head and a dependent. A dependency grammar is a set of rules that describes these dependencies. Every word (dependent) depends on another word (head), except one word which is the root of the sentence. Thus a dependency structure is a collection of dependencies for a sentence and dependency parsing depends on predicting head-modifier relationships. For example, dependency relation for the sentence "I am a student of Tezpur university". are shown in Figure 5.1. Figure 5.2 shows the dependency relation for the Assamese sentence মই নতুন কিতাপ কিনিছোঁ। (moi notun kitap kinisũ : *I have bought new book*)

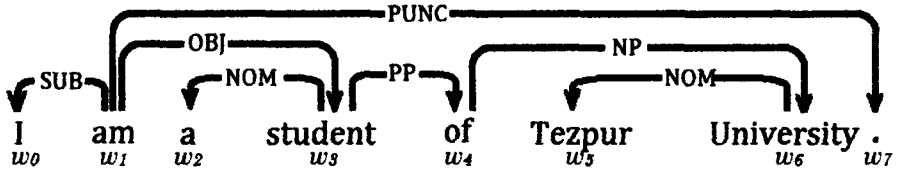


Figure 5.1: Dependency graph for sentence "I am a student of Tezpur university"

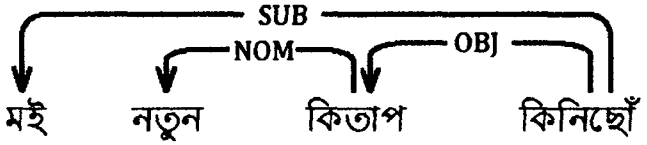


Figure 5.2: Dependency structure for the sentence "মই নতুন কিতাপ কিনিছোঁ।" (moi notun kitap kinisũ.)

Figure 5.3 shows two phrase structure analysis and Figure 5.4 shows dependency analysis for the sentence "Quickly go with dad" respectively. Dependency parsing is the process of automatically analysing the dependency structure of an input sentence. According to Nivre et al. [163], a dependency parsing model consists of a set of constraints Γ that defines the space of permissible dependency structures for a given sentence, a set of parameters λ (possibly null) and a fixed parsing algorithm h. A model is specified by $M=(\Gamma,\lambda,h)$. Thus it is the task of mapping a sentence to a well-formed dependency graph (e.g., see Figure 5.1). We can define a

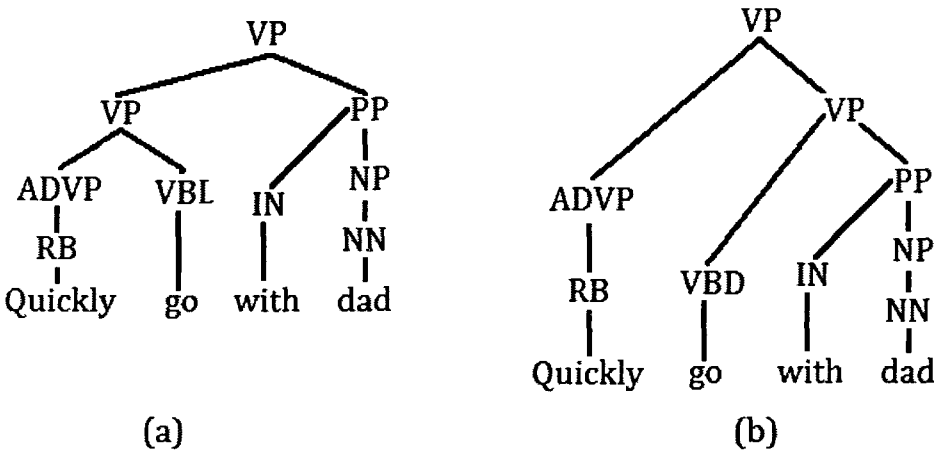


Figure 5.3: Phrase structure for the sentence "Quickly go with dad"

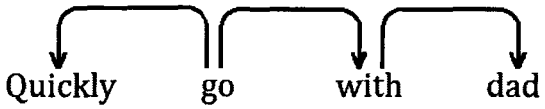


Figure 5.4: Dependency structure for the sentence "Quickly go with dad"

dependency graph G of a sentence S $(w_0, w_1, w_2, \ldots, w_n)$ over a given set R $\{r_0, r_1, r_2, \ldots, r_k\}$ of dependency types, as a labelled digraph consisting of a node set V and an arc set A, such that

1. $V \subseteq \{w_0, w_1, w_2, \ldots, w_n\}$.
2. $A \subseteq V \times R \times V$, and
3. if $(w_i, r, w_j) \in A$ then $(w_i, r', w_j) \notin A, \forall r' \neq r$.

An arc (w_i, r_k, w_j) represents a dependency relation from head w_i to dependent w_j labeled with a relation type r_k. A dependency graph G is well-formed iff

1. The node w_0 is a root of G,
2. G is connected and acyclic, and
3. Every node in G has an indegree of at most 1.

In Figure 5.1, for the sentence, *I am a student of Tezpur University*, each argument (SUB, OBJ and extension of OBJ) semantically depends on verb ***am***. If a word w_1 takes a

certain morphological form under the influence of another word w_2, w_1 morphologically depends on w_2. Dependency parsing offers some advantages.

1. Dependency links are close to the semantic relationships needed for the next stage of interpretation. For example, it is not necessary to read the head modifier for a tree that does not show it directly, for head complement relations.
2. The dependency tree contains one node per word. Because the parser's job is only to connect existing nodes, and not to postulate new ones, the task of parsing is, in some sense, more straightforward.
3. Dependency parsing lends itself to a-word-at-a-time operation, i.e., parsing by accepting and attaching words one at a time, rather than by waiting for complete phrases.

Dependency parsing algorithms can be broadly categorized into two groups. A *data-driven dependency parsing algorithm*, develops a parsing model, based on linguistic data using machine learning approaches. A *grammar-driven dependency parsing algorithm*, develops a parsing model, based on a formal grammar. A *projective graph* is one where the edges can be drawn in the plane above the sentence with no two edges crossing. On the other hand, a non-projective dependency graph does not satisfy this property. *Non-projectivity* arises due to long distance dependencies or in languages with relatively flexible word order. For many languages a significant portion of sentences require a non-projective dependency analysis [164]. In the next three sub-sections we discuss three different parsing approaches using dependency grammar.

5.3 Related work

A classifier based dependency parser, proposed by Sagae and Lavie [165], produces a constituent tree in linear time. The parser uses a basic bottom-up shift-reduce stack based parsing algorithm like Nivre and Scholz[166], but employs a classifier to determine parser actions instead of a grammar. Like other deterministic parsers (and unlike other statistical parser), this parser considers the problem of syntactic analysis separately from part-of-speech (POS) tagging. Because the parser greedily builds trees bottom-up in a single pass, considering only one path at any point in the analysis, the task of assigning POS tags to word is done before other syntactic analysis. This classifier based dependency parser shares similarities with the dependency parser of Yamada and Matsumoto [167] in that it uses a classifier to guide the parsing process in deterministic fashion, while Yamada and Matsumoto use a quadratic run-time algorithm with multiple passes over the input string.

Bharati et al. [108] describe a dependency grammar formalism called the "Paninian Grammar" Framework based on the work of Panini. Afterwards a successful application of the same framework is reported for English [168], Hindi [169], Telugu [170], Tamil [170] and Bengali [170]. The Paninian framework uses the notion of *Karaka* relations between verbs and nouns in a sentence. They suggest that the framework applied to modern Indian languages will give a new dimension to mapping syntactic and semantic relations.

A language-wise survey (see Table 5.2) shows that the Malt parser has been implemented for a variety of languages such as relatively free word order languages (e.g., Turkish), inflectionally rich languages (e.g., Hindi), fixed word order languages (e.g., English), and relatively case-less and less inflectional languages (e.g., Swedish) whereas the Paninian grammar framework has been implemented only for Indian languages and English and the CCG approach has been implemented for Dutch, Turkish and English. The other mostly implemented parsers are Collin's [171, 172] and Mc-Donald's parsers [5].

Malt Parser	English[166], Czech[173], Swedish[174], Chinese[175], Bulgarian[176], Turkish[177], Hindi[159]
Collin's Parser	English[171], Czech[178], Spanish[179], Chinese[180], German[181]
MST Parser	English[5], Czech[5], Danish[182]
CCG Framework	English[183], Dutch[184], Turkish[185]
Paninian Grammar	English[168], Hindi[168], Telugu[186], Bengali[170], Arabic[187]

Table 5.2: Language-wise survey of implemented parsers.

ICON[1]-2009 and ICON-2010[2] editions of tool contests on dependency parsing on Indian languages, provided the impetus for significant work on Indian language parsing systems after the Paninian framework was described for Indian languages. The ICON tool contests created a wave of parsing in Indian languages, at least for baseline analysis. A number of approaches were tested and analysed during that period. A transition based dependency shift-reduce parser that uses a multilayer perceptron classifier with a beam search strategy was reported by Attardi et al. [188] for Bengali, Hindi and Telugu. Ghosh et al. [189] describe a dependency parsing model for Bengali using the Malt Parser formalism. Another Malt parser based formalism was described by Kolachina et al. [190] for Hindi, Telugu and Bengali. Kesidi et al. [191] report a hybrid constraint based parser for Telugu. The next section will focus on linguistic features of Assamese.

[1]International Conference on Natural Language Processing; http://ltrc.iiit.ac in/icon/2009/nlptools
[2]http://ltrc.iiit.ac.in/icon/2010/nlptools

5.4 Assamese as a relatively free word order language

For most languages that have a major class of nouns, it is possible to define a basic word order in terms of subject(S), verb(V) and object(O). There are six theoretically possible basic word orders: SVO, SOV, VSO, VOS, OVS, and OSV. Of these six, however, only the first three normally occur as dominant orders [192]. If constituents of a sentence can occur in any order without affecting the gross meaning of the sentences (the emphasis may be affected), such types of languages are known as free word order languages. Warlpiri, Russian and Tamil are examples of free word order languages.

Typical Assamese sentences can be divided into two parts: Subject(S) and Predicate(P). The predicate may again be divided into following constituents – object(O), verb(V), extension(Ext) and connectives(Cv). A minimum sentence may consist of any one of S, O, V, Ex or Cv. Table 5.3 shows some two-constituent sentences where the words may occur in any order.

PN+V	মই আহিছোঁ। mɒi ahiso	V+PN	আহিছোঁ মই। ahiso mɒi	*I have come.*
N+V	কিতাপখন পঢ়িলোঁ। kitapk^{h}ɒn pɒrhilo	V+N	পঢ়িলোঁ কিতাপখন। pɒrhilo kitapk^{h}ɒn	*(I have) read the book.*
Adj+V	ভাল গাইছে। val gaise	V+Adj	গাইছে ভাল। gaise val	*Sang well.*
PP+V	যদি আহা! zɒdi aha	V+PP	আহা যদি! aha zɒdi	*If (you) come?*

Table 5.3: Two constituent sentences.

Assamese has a number of morpho-syntactic characteristics which make it different from other Indic languages such as Hindi. Our study reveals that word order at the clause level is free, and in some cases intra-clause level ordering is also free – that is elements which can be thought as a single semantic unit, can be reordered within the clause. The most favourite word order of Assamese is SOV. Some examples are given below.

1. মই ভাত খালোঁ। (SOV)
   (mɒi b^{h}at k^{h}alo. : *I ate rice.*)
   Now we can arrange these 3 constituents in 3! ways. Thus, we get 6 possible combinations.
   (a) ভাত খালোঁ মই। (OSV) b^{h}at k^{h}alo mɒi.
   (b) ভাত মই খালোঁ। (OVS) b^{h}at mɒi k^{h}alo.
   (c) মই খালোঁ ভাত। (SVO) mɒi k^{h}alo b^{h}at.
   (d) খালোঁ ভাত মই। (VOS) k^{h}alo b^{h}at mɒi.
   (e) খালোঁ মই ভাত। (VSO) k^{h}alo mɒi b^{h}at.

It is not necessary that all sentences have subject, verb and object. For example in the following sentence the verb is absent.

2. মই তেজপুৰ বিশ্ববিদ্যালয়ৰ ছাত্ৰ
(mɔi tezpur biswɔbidjaljɔr satrɔ : *I am student of Tezpur University*)
In this case the verb হয় (hj : *be*) is absent. Though there are 4 words, তেজপুৰ বিশ্ববিদ্যালয় (ৰ) is a single constituent, the name of a university, and so the number of constituents is 3 and hence a total of 3! grammatically correct combinations are possible. Let us consider the sentence given below.

3. মানুহজনে কুকুৰটো ৰাস্তাত দেখিছে।
(manuhzɔne kukurto ɾastat dek^{h}ise : *The man has seen the dog on the road.*)
NP - মানুহজনে (manuhzɔne : *the man*) (Man + DM)
NP - কুকুৰটো (kukurto : *the dog*) (dog + DM)
NP - ৰাস্তাত (ɾastat : *on road*) (road + LCM)
VP - দেখিছে (dek^{h}ise : *saw*) (see + past indefinite)

An interesting property of such types of sentences is that we can simply exchange the positions of the noun phrases (NP) without changing the meaning.

(a) কুকুৰটো মানুহজনে ৰাস্তাত দেখিছে।
IPA: kukurto manuhjɔne ɾastat dek^{h}ise.

(b) ৰাস্তাত মানুহজনে কুকুৰটো দেখিছে।
IPA: ɾastat manuhjɔne kukurto dek^{h}ise.

If we insert a numeral classifier এটা before NP কুকুৰ (kukur : *dog*), the total number of constituent will be increased to 5, and the sentence will be as given below.

4. মানুহজনে এটা কুকুৰ ৰাস্তাত দেখিছে।
manuhjɔne eta kukurto ɾastat dek^{h}ise : *The man has seen a dog on road.*)

In this case, we will not get 5! grammatically correct combinations. This is because the count noun এটা (eta : *one/a*) modifies only কুকুৰ (kukur : *dog*), not the others. Therefore during reordering of a sentence এটা কুকুৰ (eta kukur : *a dog*) is considered a single constituent. Sometime within the constituent, reordering of words is also possible. For example- এটা কুকুৰ (eta kukur) can be written as কুকুৰ এটা (kukur eta) without changing the meaning of the phrase. But this will introduce an ambiguity.

5.5 Parsing of Assamese text

Rahman et. al [193] proposed a method to parse Assamese text. They studied the issues related to parsing and modified the Early parser to parse simple Assamese sentences. They used only seven parts of speech tags for the words and tested over a limited amount of sentences. In the Indian languages context, PG is a well studied and already established approach. Therefore, in this work we explore other dependency-base frameworks such as Malt Parser, MST parser and Link grammar in the context of Assamese. For the link parser, we have developed a link grammar to analyse Assamese text. In Section 5.5.1 we discuss the link grammar and it's performance. We also experiment with Malt parser and MST parser.

5.5.1 Link Grammar parser

Link grammar [194] is a grammatical framework to process grammars predicated on dependency structure. In this formalism, a language is defined by a grammar that includes the words of the language and their linking requirements. Being a dependency formalism, it has the following properties.

- Connectivity: A given sentence is accepted by the system if the linking requirements of all the words in the sentence are satisfied.
- Planarity: None of the links between the words cross each other.
- Satisfaction: A set of links between the words of a sentence that is accepted by the system is called linkage. The linkage must satisfy the linking requirements of all the words.
- Exclusion: There is at most one link between any pair of words.

The grammar is defined in a file called *dictionary* and each of the linking requirements of words is expressed in terms of connectors in the dictionary file. Consider the following example.

দেউতা (deuta : *father*) : S+
বজাৰলৈ (bɒzarɔlɔi : *to market*) : O+
গ'ল (gɔl : *has gone*) : O- & S-

In the above example, S+, S-, O+ and O- are the attributes of the words for the sentence দেউতা বজাৰলৈ গ'ল (deuta bɒzarɔlɔi gɔl), which represent computer readable encoding for expressing the linking requisites. The + or - symbol indicates the direction from the head to the dependent or vice verse. Letters preceded by + or - symbol are the names of the connectors. Thus the linking requirements for each word consist of connector names followed by directions of the links, parentheses to denote precedence and & and OR operators that handle more than one outgoing

or incoming links from the word (if any). The word দেউতা (deuta) has an S+ connector. To establish a link, it requires a words with an S- connector. The word গ'ল (gɔl) has an S- connector and establishes a linkage from দেউতা (deuta) to গ'ল (gɔl) with a connection label S (Subject). This is the working principle of link grammar. For our work, we have written a dictionary file of around two hundred definitions that include noun-verb agreement, various nominal modifiers, clauses and questions. There are a number of common phenomena that we are not able to handle by the current definitions, that include complex clauses and serial verbs. Appendix C shows the structure of our dictionary file and definitions for a link grammar for Assamese. The current grammar can parse simple, complex and compound sentences. The following figures [5.6–5.8] shows the dependency structure for some simple and complex sentences.

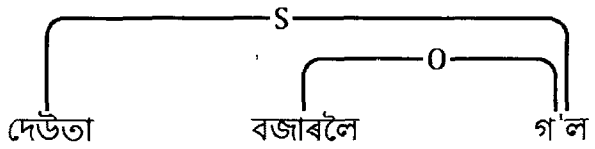


Figure 5.5: Dependency graph for the simple sentence দেউতা বজাৰলৈ গ'ল (deuta bɒzarɔlɔi gɔl)

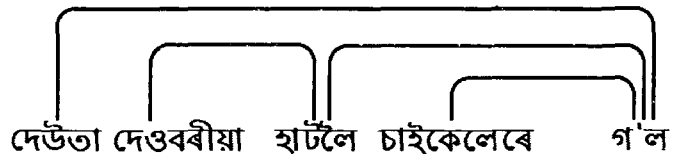


Figure 5.6: Dependency graph for the sentence দেউতা দেওবৰীয়া হাটলৈ চাইকেলেৰে গ'ল (deuta deobɒrija hatɒloi saikelere gɔl)

Figure 5.7: Dependency graph for the sentence দেউতাই ৰঙা কামিজটো পিন্ধি দেওবৰীয়া হাটলৈ গ'ল (deutai rɒŋa kamizto pind^{h}i deobɒrija hatɒloi gɔl)

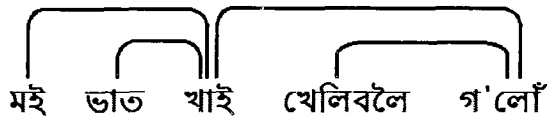


Figure 5.8: Dependency graph for the sentence মই ভাত খাই খেলিবলৈ গ'লোঁ (mɔi b^{h}at k^{h}ai k^{h}ɛlibɔlɔi gɔlo)

In Figure 5.9, the connector "RB-" indicates that the adjective to be modified by an adverb on its left. This connector is conjoined with the "NoM+" connector to indicate that

```
NoM :(<affix> & ((RB- & NoM+) or ([<ns-1-right>])))
RB: (<affix> & (EE- & (EE+ or RB+))) or <rb-2-vb>;
<noun-GEN-right>:(Dg+ or RPg+);
<noun-DAT-right>:(IOd+ or RPd+);
<noun-ACC-right>:(Oc+);
<noun-LOC-right>:(IOl+);
```

Figure 5.9: Linking requirements of nominal modifiers, adverbs and nouns

an adjective can modify a noun on the right hand side. In addition, these two connectors are disconnected from with the "<ns-1-right>" macro on the right hand side. In fact, this macro is for the syntactic role of nouns as a modifier. Disconnection of this macro from the existing formula of adjectives enables them to behave as nouns. The following sentence illustrates the usage.

ধুনীয়া ছোৱালীজনী গ'ল। (dʰunia sowalizɒni gɔl)

In this example, the word ছোৱালীজনী (sowalizɒni : *the girl*) is the subject of the verb গ'ল (gɔl : *go+present perfect*); hence, it is connected with an "Sdm", subject (S) with definitive marker (dm) link. The first word ধুনীয়া (dʰunia : *beautiful*) modifies the subject noun, ছোৱালীজনী (sowalizɒni : *the girl*). A noun modifies the verb either as a subject or as an object. A nominal group consists of a group of nouns that are connected to each other with some relations. In a nominal group, a word can be modified by more than one word. Both the modified item and modifiers themselves belong to the nominal group.

5.5.1.1 Links and performance analysis

We tested the performance of the system for coverage with 100 randomly selected sentences from tagged corpus, of which the first ten sentences are given in Table 5.4. The average number of words in the sentences is 5.82. The average number of parses per sentences is 3.21. Among 100 parsed sentences, 69 sentences contain the correct parse. This shows that, though there exist some issues out of our scope, we handle most of the important phenomena.

5.5.2 Malt Parser

The Malt parser described in [166, 4, 161] is a deterministic and classifier based dependency parser. This inductive dependency parsing methodology [195] requires three essential compo-

Sl.	Sentence	A	B	C
1	দেউতা বজাৰলৈ গ'ল।	3	1	2
3	দেউতাই বজাৰৰ পৰা নতুন কাপোৰ আনিছে।	6	1	4
4	গৰু এবিধ ঘৰচীয়া জন্তু।	3	1	2
5	মা আৰু দেউতা বজাৰলৈ গ'ল।	5	1	2
6	প্ৰায় সকলো মানুহৰে সুন্দৰৰ প্ৰতি আকৰ্ষণ আছে।	7	1	5
7	সকলোৱে নিজক সজাবলৈ ঘৰখন বা বাৰীখন ধুনীয়াকৈ ৰাখিবলৈ বিচাৰে।	9	0	7
8	প্ৰয়োজনত মানুহে কাপোৰ পিন্ধিলেও কেৱল প্ৰয়োজন দূৰ কৰাতে মানুহ সন্তুষ্ট নহয়।	11	0	6
9	মই ভাত খালোঁ আৰু খেলিবলৈ গলোঁ।	6	1	3
10	এই সৌন্দৰ্য্যবোধৰ কাৰণেই মানুহৰ মাজত শিল্পী সাহিত্যিকৰ সৃষ্টি হৈছে।	9	0	5
11	মই ভাত খালোঁ আৰু খেলিবলৈ গলোঁ।	6	1	3

Here, A – Number of words per sentence; B – Does the resulting parse set contain the correct parse? (1 if yes and 0 otherwise); C – number of possible parses found in the sentence.

Table 5.4: Statistics of the input sentences for performance evaluation

nents, given below.

1. A deterministic parsing algorithm used to build dependency graphs,
2. History based feature models for predicting the next parser action, and
3. A machine learning technique to decide the parser action based on histories.

Researchers have used many different approaches for each of the three components. To create a dependency graph for a given sentence, the parser uses two data types. A *stack* is used for partially processed tokens, and a *list* holds the tokens still to be processed. At any given time during the parse, the parser has three actions available to it.

1. The first is a shift, where the next token is pushed onto the stack.
2. The second and third are to create a left dependency arc and a right dependency arc, respectively. In the left-arc operation, a dependency arc is drawn from the next token to the token at the top of the stack and the top token is popped off.
3. The right-arc operation draws a dependency arc from the top token of the stack to the next token in the list, and replaces the head of the list with the top token of the stack.

5.5.2.1 Results and analysis

For our experiment we use the freely available Malt Parser[3]. The performance of the parser is dependent on three parameters, viz., parameter for parsing algorithms, parameter for learning algorithms and feature parameter. For most of the cases, we use default parameters values. Malt parser version 1.7.2 comes with two learning algorithms, viz. LibSVM [196] and LibLINEAR [197]. As this is the first kind of work for Assamese, we restricted our experiments on arc-eager and arc-standard mode only. We prepare the input in the form of CoNLL[4] shared task format. Statistics for training and testing dataset are given in Table 5.5.

	Sentences	Words	Avg. words	Simple	Compound	Complex
Training	200	1,228	6.14	120	50	30
Testing	80	487	6.08	50	20	10

Table 5.5: Statistics of used corpus for training and testing Malt parser

To calculate the average accuracy, we set the cost parameter of the learning algorithm that controls the boundary between maximizing training error and maximizing margin to 0.70. The average labeled attachment score (LAS), unlabeled attachment score (UAS) and labeled score (LS) on the test dataset are given in Table 5.6. It is seen that due to the very small dataset, the dependency relations are sparse, which results the low LAS value.

Algorithm	Classifier	LAS	UAS	LS
Arc Eager	LibSVM	61.12	50.45	54.20
Arc Eager	LibLinear	61.80	51.15	62.00
Arc Standard	LibSVM	62.75	50.30	53.62
Arc Standard	LibLinear	63.80	50.52	55.85

Table 5.6: Average accuracy obtained using Malt parser

5.5.3 MST Parser

This approach views dependency structures as maximum spanning trees (MST). It furnishes a general framework for parsing trees in both projective and non-projective sentences. The MST parser obtains dependency relations among constituents in a sentence by searching for maximum spanning trees in directed graphs [5, 182]. A maximum spanning tree for a given sentence is defined as the tree with the greatest score out of all the possible parse trees. In this parsing

[3]Malt parser version 1.7.2; http://www.maltparser.org. Access date: 1 December 2012

[4]Conference on Natural Language Learning Shared Task 2006; http://ilk.uvt.nl/conll/; Access date:1 December 2012

model, dependency is analysed as a problem of finding a maximum spanning tree. Each sentence is modelled as a digraph, where each word is connected to every other by a scored directed edge. An extra node in the graph is included to serve as the "root" node, whose dependent is the root of the sentence. Initially this node is connected to every word in the sentence through a directed edge and an associated score. Scores for each edge are determined by a scoring function, which is defined as the dot product of a weight vector, and a high dimensional feature representation.

5.5.3.1 Results and analysis

For our experiment, we use the freely downloadable MST parser[5] [198], version 0.2. For training and testing, we use the same dataset used for the Malt parser. Each sentence in the training dataset is represented by 4 lines and words of a sentence are space separated. The general format is:

w_1 w_2 w_3 ... w_n
p_1 p_2 p_3 ... p_n
l_1 l_2 l_3 ... l_n
d_1 d_2 d_3 ... d_n

where,
$w_1 \ldots w_n$ are the n words of the sentence (tab delimited),
$p_1 \ldots p_n$ are the POS tags for each word,
$l_1 \ldots l_n$ are the labels of the incoming edge to each word, and
$d_1 \ldots d_n$ are integers representing the position of each words parent.

The average labeled attachment score (LAS), unlabeled attachment score (UAS) and labeled score (LS) on the test dataset are given in Table 5.7.

LAS	UAS	LS
61.12	50.45	54.20

Table 5.7: Average accuracy obtained using the MST parser

Our aim in these experiments is to explore the different possibilities of parsing Assamese text and we have in a way, achieved the goal, although the size of the dataset is small for data driven parsing approaches like the Malt and MST. The results obtained from the three approaches are summarized in Table 5.8. As expected, being a grammar based approach, the Link grammar parser shows better accuracy than the other two parsing approaches.

[5]http://www.seas.upenn.edu/ strctlrn/MSTParser/MSTParser.html. Access date: 12 January 2013.

Formalism	LAS	UAS	LS
Link parser	78.80	75.15	72.00
Malt parser	61.80	51.15	62.00
MST parser	61.12	50.45	54.20

Table 5.8: Parsing results using Link grammar, Malt parser and MST parser.

In Malt parsing (transition based deterministic parsing), if the transition function can be computed in constant time, the parsing algorithm takes $O(n)$ time in the best case. Thus it is possible to parse a sentence in linear time. If the parse is required to be projective, it takes $O(n^2)$ time. With MST projective dependency parsing, the Cocke-Kasami-Younger (CKY) algorithm, a context free grammar parsing approach, is used, taking $O(n^5)$ time. The same job performed with Eisner's bottom-up-span algorithm [183] reduces the time to $O(n^3)$. On the other hand, for non-projective parsing, the parser uses the CLE algorithm for constructing the MST, taking $O(n^2)$ time. The Eisner and the CLE algorithms are quite distinct from each other. The Eisner algorithm uses bottom-up dynamic programming, but the CLE algorithm is greedy recursive. Since the approach is greedy recursive, the time complexity is less than Eisner's bottom-up dynamic approach.

5.6 Experiments in other languages

In this thesis we consider "Assamese" as a representative of the considered set of languages and experiment with three important problems of natural language understanding namely stemming, PoS tagging and Parsing. We also reported that the stemming and PoS tagging approaches considered for Assamese are giving acceptable results for languages like Bengali, Bodo and Bishnupriya Manipuri (Section 3.7 and Section 4.8). So we can speculate an acceptable accuracy for other languages in parsing. Our Analysis shows the truth of the speculation. Let us consider two sentences from each languages namely Bodo (member of Tibeto-Burman language family) and Bishnupriya Manipuri (member of Indo-Aryan language family).

Sentence 1:
Assamese : মই ভাত খাই খেলিবলৈ গ'লো। (moi bʰat kʰai kʰelibɔlɔi gɔlo)
Bodo : আং আঁখাম জাখানানৈ গেলেনো থাঙো। (aŋ ɯŋkʰam zakʰaŋnanɯi gɛlɛnɯ tʰaŋɯ)
Bishnupriya Manipuri : মি ভাত খেয়া খেলানিত গেলুগা। (mi bʰat kʰeja kʰelanit gɛluga)
Sentence 2:
Assamese : দেউতা দেওবৰীয়া হাটলৈ চাইকেলেৰে গ'ল। (deuta deobɒrija hatɒloi saikelere gɔl)
Bodo : আফায়া দেবৱার হাথাযাব সাইখেলজাঁ থাংবায়। (apʰaja dɛobar hatʰajaw sajkʰɛlzuŋ tʰaŋbaj)
Bishnupriya Manipuri : বাবা লাম্বাইছিঙর বাজারে চাইকেলহানল' গেলগা। (baba lambajsiŋɒr bazare saikɛlhanlɔ gɛlga)

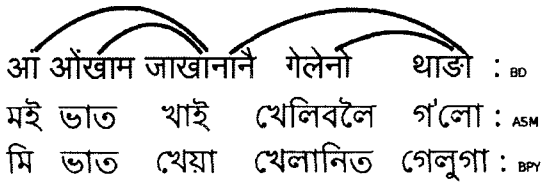


Figure 5.10: Dependency graph for sentence 1.

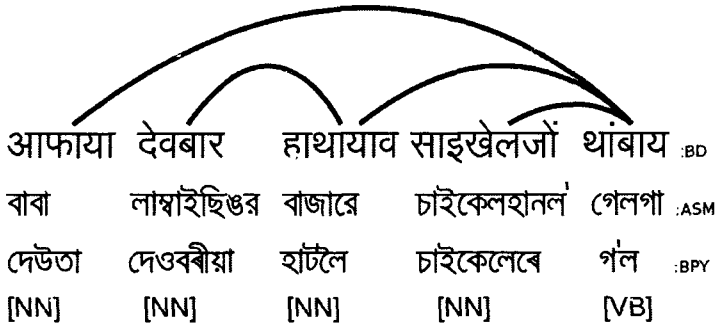


Figure 5.11: Dependency graph for sentence 2.

From the dependency graph shown above, it is seen that irrespective of the considered languages, functional components are similar at least for few simple analysed sentences. On the basis of the analysis, we may conclude that the findings for Assamese can be applied to Bodo and Bishnupriya Manipuri sentences with added linguistics details. To develop a complete parser for these languages very first thing we need is linguistic experts and resources. Developing a relation-extractor for resource-poor and expertise-poor languages will be an interesting future work.

5.7 tezuBank: Assamese Dependency TreeBank[6]

A TreeBank or a parsed corpus is a text corpus in which each sentence has been annotated with syntactic structure. The syntactic structure is commonly represented as a tree structure, hence the name TreeBank. The term parsed corpus is often used interchangeably with TreeBank: with the emphasis on the primacy of sentences rather than trees. It is widely admitted that Tree-Banks constitute a crucial resource both to develop NLP applications and to acquire linguistic knowledge. The use of annotated corpora may lead to rapid progress in text understanding and

[6]Freely downloadable at www.tezu.ernet.in/~nlp/res.htm

spoken language understanding systems, which in turn may lead to faster development of automatic information extraction systems. Thus, such corpora are important research tools for text processing, speech recognition, psycholinguistic study and theoretical linguistics. In this section, we describe the building of an Assamese TreeBank from/for the parsing approaches described in the previous sections. TreeBanks can be used in corpus linguistics for studying syntactic phenomena or in computational linguistics for training or testing parsers. Once completely parsed, a corpus contains evidence of both frequency (how common different grammatical structures are) and coverage (the discovery of new and unanticipated grammatical phenomena).

Rambow et al. [199] propose dependency rather than phrase-structure annotation for English. The Turin University TreeBank (TUT)[7] [200] for Italian consist of 2860 sentences with morphology, syntax and semantic annotations. They also have developed TUT-Penn converter [201], which convert TUT annotation format to Penn TreeBank annotation and TUT-CCG converter [202], which convert TUT annotation format to CCG format. The Prague Dependency TreeBank (version 2.5) for Czech [203] and Arabic [204] have four annotation layers: word layer, morphological layer, analytical layer and tectogrammatical layer. They use XML-based annotation format called PML (Prague Markup Language) for annotation. In Arabic TreeBank version 2.0, 383482, 282252, 30894 words are annotated and interlinked with morphological, analytical and tectogramatical annotation respectively; while 2, 1.5 and 0.8 million words for Czech are annotated and interlinked with morphological, analytical and tectogramatical layer respectively. The Prague Czech-English Dependency TreeBank [205] has 50000 sentences each of Czech and English from a Czech-English parallel corpus. The parallel corpus is created by translating English texts from Penn TreeBank to Czech. The METU-Sabanci Turkish TreeBank[8] is a collection of 7262 sentences with XML-based morphology and syntactic annotation for Turkish. The sentences are collected from XCES (Corpus Encoding Standard for XML) [206] tagged METU corpus. The Copenhagen Dependency TreeBank (CDT)[9] [207] for Danish contains 100000 words and Danish-English parallel corpus contains 95000 words with morphology and syntax annotation. Other major efforts in the dependency framework are Chinese Dependency TreeBank [10] [208] for Chinese, Alpino TreeBank[11] [209] for Dutch, Turku Dependency TreeBank (TDT)[12] [210] for Finnish, TIGER [211] that combines dependency with PSG for German.

The first attempt to develop such a resource for Indian languages was taken up at the "Workshop on Lexical Resources for Natural Language Processing, 2001", held at IIIT Hyderabad. The task was named as AnnCorra, shortened for "Annotated Corpora". They uses

[7]Turin University TreeBank; http://www.di.unito.it/~tutreeb
[8]METU-Sabanci Turkish TreeBank; http://ii.metu.edu.tr/corpus
[9]https://code.google.com/p/copenhagen-dependency-TreeBank
[10]Chinese Dependency TreeBank; http://catalog.ldc.upenn.edu/LDC2012T05
[11]Alpino TreeBank; http://www.let.rug.nl/vannoord/trees
[12]Turku Dependency TreeBank; http://bionlp.utu.fi/finTreeBank.html

SSF (Shakti Standard Format) [212] for annotation. Since Indian languages are morphologically rich and have relatively flexible word order, for achieving this, certain standards had to be drawn in terms of selecting a grammatical model and developing tagging schemes for the three levels of sentential analysis: PoS tagging (lexical level), chunking (phrase level) and syntactic parsing (sentence level). The annotation scheme proposed by Begum et al. [213] based on Paninian framework reports preliminary experiments with 1403 Hindi sentences.

Since there is no formal syntactic description of Assamese yet, we are free to choose any approach we find appropriate. We have seen that context free grammars are not well suited for free word order languages such as Assamese. Instead, the dependency framework appears to be better suited. Also the dependency framework is arguably closer to semantics than Phrase Structure Grammar [213]. While the constituency annotation was the system used in the first TreeBank project, the dependency annotation has become more popular in last decads as the number of multilingual TreeBanks has increased. In TreeBanks the constituency based annotation schemes are motivated by underlying generative formalisms describing the hierarchy and composition of constituents (such as S $\rightarrow$ NP VP) in a sentence. The dependency based annotation schemes are motivated by underlying dependency formalisms trying to define dependency relations between parts of the sentence (such as `eat (the_cat, the_rat)`). Each approach has its pros and cons and probably the best solution would be to record both annotations for each sentence in the TreeBank. The primary reasons for using dependency structures instead of more informative lexicalized phrase structures is that they are more efficient to learn and parse while still encoding much of the predicate-argument information needed in applications [5].

In this work, we start the initial roadmap to build a large text corpus and named it as tezuTreeBank, shortening the full name "Tezpur University Assamese TreeBank". This research aims to develop a fully annotated large scale Assamese TreeBank. It is the first attempt of its kind for Assamese. As an initial task, we design a stemmer to find the root, although a morphological analyser may better perform morphological analysis of the corpus. For grammatical categorization, we develop a part-of-speech tagger to tag the corpus. Next, we develop a dependency analyser module that can parse the sentences in the corpus using dependency tag-sets, which are already described in the previous chapters and sections. We will perform manual testing and evaluation of these modules.

The creation of a TreeBank comprises several important phases: design phase, implementation phase, validation phase and documentation phase. During the design phase, two main issues are considered: the width and depth of linguistics knowledge, and the TreeBank format. The design phase describes the underlying annotation scheme for the TreeBank. The main task in the implementation phase are : identifying the source information such as text corpora, grammar formalism used, lexicons, general linguistic analysis and the tools used. The validation phase requires consistency checks over the whole TreeBank again and again. This

current version contains 200 sentences taken from the corpus described in Chapter 2, with complete morpho-syntactic analysis. This analysis was performed specifically by manual effort. We define the following morpho-syntactic tags for annotation.

- Part of speech (PoS)
- Sub-categorization
- Inflection
- Lemma (canonical form)
- Parts (with similar morphosyntactic tags) for compounds.

For part of speech annotation, we use our hierarchical tagset. A sentence from the TreeBank is given below.

< Sentence id = "50"/>
< w compound = "yes" word = "জোনবিল মেলা" phr = "NN"/>
    < w id = "1" word = "জোনবিল" lemma = "জোনবিল" pos = "NC" mrph = ""/>
    < w id = "2" word = "মেলাই" lemma = "মেলা" pos = "NC" mrph = ""/>
< w id = "3" word = "পাহাৰ" lemma = "পাহাৰ" pos = "NC" mrph = ""/>
< w id = "4" word = "আৰু" lemma = "আৰু" pos = "CCD" mrph= ""/>
< w id = "5" word = "ভৈয়ামৰ" lemma = "ভৈয়াম" pos = "NC" mrph = ""/>
< w id = "6" word = "লোকসকলৰ" lemma = "লোক" pos = "NC" mrph = ""/>
< w id = "7" word = "মাজত" lemma = "মাজ" pos = "AMN" mrph = ""/>
< w compound = "yes" word = "একতাৰ এনাজৰী" phr = "NN"/>
    < w id = "8" word = "একতাৰ" lemma = "একতা" pos = "NC" mrph= ""/>
    < w id = "9" word = "এনাজৰী" lemma = "এনাজৰী" pos = "NC" mrph = ""/>
< w id = "10" word = "সুদৃঢ়" lemma = "সুদৃঢ়" pos = "JJ" mrph = ""/>
< w id = "11" word = "কৰাত" lemma = "কৰ" pos = "VM" mrph = ""/>
< w id = "12" word = "অৰিহণা" lemma = "অৰিহণা" pos = "NC" mrph = ""/>
< w compound = "yes" word = "যোগাই আহিছে" phr = "VB"/>
< w id = "13" word = "যোগাই" lemma = "যোগ" pos = "VB" mrph = ""/>
< w id = "14" word = "আহিছে" lemma = "আহ" pos = "VB" mrph = ""/>
< w id = "15" word = "।" lemma = "।" pos = "PUN" mrph = ""/>
< /Sentence>

5.8 Summary

In this chapter, we describe the state of the art of Indian language parsing. We also explain the developed link grammar, Malt parser and MST parser for Assamese. After that we compare three widely used dependency parsing algorithms. The Malt parser is a data driven parsing approach, where the actual parsing model is prepared depending on the behavior of TreeBank/corpus data. So for languages which do not have a large amount of TreeBank data, a Link grammar parser and MST an parser may be more suitable, though computational complexity of these approaches is higher. We also experiment with other resource-poor languages such as Bodo and Bishnupriya Manipur. We describe the basic architecture of tezuBank, the Assamese dependency TreeBank at the end. We can extend our work in three basic directions. First, increase the size of TreeBank to mark it as a large annotated resource for Assamese as well as for Bodo and Bishnupriya Manipuri. Second, as considered languages are from same region and share some common vocabulary, our work can be extended to develop a parallel TreeBank. Third, we can experiment and compare with other evaluation measures such as tree similarity and PoS assignment accuracy.

Chapter 6

Conclusion

..... "We cannot say that if a child is badly nourished he will become a criminal. We must see what conclusion the child has drawn.".....

–Alfred Adler (1870 - 1937)

For Computational processing of natural languages, appropriate modelling of morphology and syntax are two primary tasks. While even for well studied languages this continues to be an active area of research, for many languages this work has barely started. In this work, we have taken up the case of Assamese, one of the several Indian languages which have received little attraction of computational linguistic research. We have successfully implemented the following.

- Stemmer: We implement three methods of stemming; viz. suffix stripping, dictionary look-up and a hybrid approach using HMM for four morphologically rich, agglutinating and relatively free word order Indian languages viz. Assamese, Bengali, Bishnupriya Manipuri and Bodo. We examine the prose and cons of each approach and reported the accuracy accordingly. We found that for the language set, a dominant fraction of suffixes are single letter and words ending such single letters create problems during suffix stripping. Therefore, we propose a new method that combines the rule-based algorithm for predicting multiple letter suffixes and an HMM-based algorithm for predicting the single letter suffixes. The resulting algorithm uses the strengths of both algorithms leading to a much higher accuracy of 94% compared to just 82% for Assamese and 94%, 87% and 82% for Bengali, Bishnupriya Manipuri and Bodo, respectively. We find that the performance of hybrid approach is encouraging for resource-poor inflectional languages.

- POS tagger: We have achieved fairly good results for PoS tagging. Firstly, we implement a suffix based noun and verb tagging approach for Assamese. We obtain around 94% accuracy in identifying nouns and verbs when testing with texts containing more than 5,000 inflected noun and 5,000 inflected verbs. Except noun and verb, other grammatical categories are closed class and relatively small like most languages. As the second task, we extend the noun and verb identification approach to PoS tagging using an ambiguous and frequent word list. We obtain around 91% accuracy for the dictionary based approach with a word list of around 10456 words. After increasing the dictionary size to 30000, we achieve 95% precision for the same. We implement an HMM based tagger as the third task and get around 87% precision with a approximately 10000 training words. As the first work on Assamese, the accuracy looks good, in comparison with work in other languages. We also study the different issues related to MWUs. We achieve 95.82%, 81.12%, 72.54% and 74.22% accuracy for reduplications, compound nouns, compound verbs and conjunct verbs respectively

  Our main achievement is the development of taggers with 87%-95% precision and creation of the Assamese tagsets. We implement an existing method for PoS tagging, but our work is for a new language where an annotated corpus and a pre-defined tagset were not available. As the word order of Assamese is relatively free, we cannot use positional information like in fixed word order languages. Another important observation from this experiment is that though Assamese is relatively free word order, some parts of speech do not occur in the initial or final positions in a sentence. We believe that embedding

linguistic word agreement rules in tagging will improve the performance of the method.

- Parser: We have developed the rule base and the dictionary for link garmmar parser for Assamese. We also tested the tagged corpus with Malt and MST parser. As obvious we find that the accuracy of link grammar parser is better than the other two. Since Malt parser is a data driven parsing approach, where the actual parsing model is prepared depending on the behavior of treebank/corpus data. So for languages which do not have a large amount of treebank data, Link grammar parser and MST parser may be more suitable, though computational complexity of these approaches is higher.

All evaluations in this work are manual. Our work in each of the languages has been evaluated by one highly educated and native speaker of the language. We compare our result with other similar reported work. Since the data sets in the reported work and the dataset we use for the experiments are different, the comparison is not absolute. Nevertheless, being the first work of its kind for Assamese and other two languages – Bodo and Bishnupriya Manipuri (The latter two are vulnerable languages according to UNESCO), the work has great impact. In conclusion we claim that we have created stemmer, PoS tagger and Link grammar parser for Assamese, a less frequently discussed and resource-poor language in computational linguistic domain.

Due to the lack of IR system, gold standard data and other resources for Assamese, we have not evaluated the impact of our proposed approaches. Our work can be extended to evaluate all the proposed approaches with some widely accepted measure like Mean Average Precision for stemming, n-gram based evaluation for PoS tagging and tree similarity or PoS assignment accuracy for parsing. As future work, it would be interesting to explore the possibility of modelling all morphological and syntactical phenomena using other successful techniques such as Optimality Theory [65], Maximum Entropy Models [66] and Conditional Random Fields [67] and comparing the results with those of our approaches.

Contribution

Published Paper

- [Journal] **Navanath Saharia**, Utpal Sharma and Jugal Kalita, "*Stemming resource-poor Indian languages*", ACM Transactions on Asian Language Information Processing, 13(3), Pages 14:1-14:26, 2014.

- [Conference] **Navanath Saharia**, Kishori M. Konwar, Utpal Sharma and Jugal Kalita, "*An Improved Stemming Approach Using HMM for a Highly Inflectional Language*", in proceedings of 14^{th} International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-13), Pages 164-173, Samos, Greece, 2013, Springer.

- [Conference] **Navanath Saharia**, Utpal Sharma and Jugal Kalita, "*Analysis and Evaluation of Stemming Algorithms: A case study with Assamese*", in proceedings of the International Conference on Advances in Computing, Communications and Informatics, Pages 842-846, Chennai, India, 2012, ACM.

- [Book Chapter] **Navanath Saharia**, Utpal Sharma and Jugal Kalita, "*A Dictionary Based POS Tagger for Morphologically Rich Language*", Machine Intelligence:Recent Advances, Narosa Publishing House, Editors. B. Nath, U. Sharma and D.K. Bhattacharyya, ISBN-978-81-8487-140-1, 2011.

- [Journal] **Navanath Saharia**. Utpal Sharma and Jugal Kalita, "*A First Step Towards Parsing of Assamese Text*", Language In India, Volume 11 (5), 2011. (National Seminar on Lexical Resource and Computational Linguistics of Indian Languages, Pondicherry, India, 2010).

- [Conference] **Navanath Saharia**, Utpal Sharma, Jugal Kalita, "*Suffix Based Noun and Verb Classifier for Assamese*", In proceedings of International Conference of Asian Language Processing, Harbin, China, 2010, IEEE.

✍ [Book Chapter] **Navanath Saharia**, Utpal Sharma, Jugal Kalita, "*A Review of Three Parsing Algorithms*", Algorithms in Applications, Narosa Publishing House, Editors. U. Sharma and D.K. Bhattacharyya, ISBN-978-81-8487-082-4, 2010.

✍ [Conference] **Navanath Saharia**, Utpal Sharma "*Parsing of Assamese Sentences*", 54^{th} Annual Technical Session of Assam Science Society, Tezpur, India, 2009.

✍ [Conference] **Navanath Saharia**, Dhrubjyoti Das, Utpal Sharma, Jugal Kalita "*Part of Speech Tagger for Assamese Text*", In proceedings of Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP, Short paper), pages 33-36, Suntec, Singapore, 2009, ACL.

Tools developed

1. *Stemmer* for Assamese, Bengali, Bishnupriya Manipuri and Bodo text.
2. *Part of Speech Tagger* for Assamese, Bengali, Bishnupriya Manipuri and Bodo text.
3. *MWE extractor* for Assamese.
4. *tezuBank* : a TreeBank for Assamese.
5. *Geetanjali (Ramdhenu) to Unicode Converter*: A tool to convert Geetanjali font (A propeitory font, mostly used in print-houses to print news papers, books, and novels) text to Unicode Assamese.
   [Freely downloadable at www.tezu.ernet.in/~nlp/g2u.htm]
6. *Roman to Unicode Converter*: A tool to convert Assamese text written using Roman script (written like Mobile SMS) to Unicode.
   [Freely downloadable at www.tezu.ernet.in/~nlp/r2u.htm]

Appendix A

Assamese noun and verb inflections

We found 5,260 inflectional forms for the Assamese noun root মানুহ (manuh : *man*). Table A.1 lists a few of the obtained inflectional forms. The second column is the inflected word, third column represents the transcribed form in IPA and the fourth column shows the break-up of the word into valid morphological units.

Sl.	Word	IPA	Root+Suffixes
1.	মানুহৰ	manuhɒr	মানুহ+ৰ
2.	মানুহহে	manuhhɛ	মানুহ+হে
3.	মানুহটো	manuhto	মানুহ+টো
4.	মানুহেনো	manuhɛno	মানুহ+ে◌+নো
5.	মানুহেহে	manuhɛhɛ	মানুহ+ে◌+হে
6.	মানুহেইতো	manuhɛito	মানুহ+ে◌+ইতো
7.	মানুহেও	manuhɛo	মানুহ+ে◌+ও
8.	মানুহেওনে	manuhɛonɛ	মানুহ+ে◌+ও+নে
9.	মানুহেওচোন	manuhɛoson	মানুহ+ে◌+ও+চোন
10.	মানুহজন	manuhʤɒn	মানুহ+জন
11.	মানুহজনক	manuhʤɒnɒk	মানুহ+জন+ক
12.	মানুহজনৰ	manuhʤɒnɒr	মানুহ+জন+ৰ
13	মানুহজনলৈ	manuhʤɒnɒloi	মানুহ+জন+লৈ
14.	মানুহজনত	manuhʤɒnɒt	মানুহ+জন+ত
15.	মানুহজনী	manuhʤɒni	মানুহ+জনী
16.	মানুহজনীক	manuhʤɒnik	মানুহ+জনী+ক
17.	মানুহজনীৰ	manuhʤɒnir	মানুহ+জনী+ৰ
18.	মানুহজনীলৈ	manuhʤɒniloi	মানুহ+জনী+লৈ
19.	মানুহজনীত	manuhʤɒnit	মানুহ+জনী+ত
20.	মানুহজনেৱেতো	manuhʤɒnɛrɛto	মানুহ+জন+◌েৱে+তো
21.	মানুহজনেৱেচোন	manuhʤɒnɛrɛson	মানুহ+জন+◌েৱে+চোন
22.	মানুহটোৱে	manuhtorɛ	মানুহ+টো+ৱে

23.	মানুহটোৰবা	manuhtoɒrba	মানুহ+টো+ৰ+বা
24.	মানুহমখাৰহে	manuhmɒkʰaɒrhe	মানুহ+মখা+ৰ+হে
25.	মানুহমখাৰহেনে	manuhmɒkʰaɒrhene	মানুহ+মখা+ৰ+হে+নে
26.	মানুহবিলাকলৈও	manuhbilakloio	মানুহ+বিলাক+লৈ+ও
27.	মানুহবিলাকলৈওচোন	manuhbilakloioson	মানুহ+বিলাক+লৈ+ও+চোন
28.	মানুহবোৰ	manuhbor	মানুহ+বোৰ
29.	মানুহবোৰৰে	manuhboɾe	মানুহ+বোৰ+ৰে
30.	মানুহবোৰৰেও	manuhboɾeo	মানুহ+বোৰ+ৰে+ও
31.	মানুহবোৰৰেওচোন	manuhboɾeoson	মানুহ+বোৰ+ৰে+ও+চোন
32.	মানুহকেইজনক	manuhkeiʤɒnɒk	মানুহ+কেইজন+ক
33.	মানুহকেইজনকনে	manuhkeiʤɒnɒkne	মানুহ+কেইজন+ক+নে
34.	মানুহকেইজনৰচোন	manuhkeiʤɒnɒrson	মানুহ+কেইজন+ৰ+চোন
35.	মানুহকেইজনীমানক	manuhkeiʤɒnimanɒk	মানুহ+কেইজনী+মান+ক
36	মানুহকেইজনীমানৰ	manuhkeiʤɒnimanɒr	মানুহ+কেইজনী+মান+ৰ
37	মানুহকেইজনীমানলৈ	manuhkeiʤɒnimanɒloi	মানুহ+কেইজনী+মান+লৈ
38.	মানুহকেইজনমানেৰে	manuhkeiʤɒnmaneɾe	মানুহ+কেইজন+মান+েৰে
39.	মানুহকেইজনমানেৰেইনো	manuhkeiʤɒnmaneɾeino	মানুহ+কেইজন+মান+েৰে+ইনো
40.	মানুহকেইজনমানেৰেইহে	manuhkeiʤɒnmaneɾeihe	মানুহ+কেইজন+মান+েৰে+ইহে
41.	মানুহকেইজনমানেৰেইহেচোন	manuhkeiʤɒnmaneɾeiheson	মানুহ+কেইজন+মান+েৰে+ইহে+চোন
42.	মানুহকেইজনমানেৰেও	manuhkeiʤɒnmaneɾeo	মানুহ+কেইজন+মান+েৰে+ও
43.	মানুহকেইজনমানেৰেওনে	manuhkeiʤɒnmaneɾeone	মানুহ+কেইজন+মান+েৰে+ও+নে
44.	মানুহকেইগৰাকীমানেৰেহে	manuhkeigɒrakimaneɾehe	মানুহ+কেইগৰাকী+মান+েৰে+হে
45	মানুহকেইগৰাকীমানেৰেহেনে	manuhkeigɒrakimaneɾehene	মানুহ+কেইগৰাকী+মান+েৰে+হে+নে
46.	মানুহকেইগৰাকীমানেৰেহেচোন	manuhkeigɒrakimaneɾehe∫on	মানুহ+কেইগৰাকী+মান+েৰে+হে+চোন
47.	মানুহকেইগৰাকীমানেৰেবা	manuhkeigɒrakimaneɾeba	মানুহ+কেইগৰাকী+মান+েৰে+বা
48.	মানুহকেইগৰাকীমানেৰেতো	manuhkeigɒrakimaneɾeto	মানুহ+কেইগৰাকী+মান+েৰে+তো
49.	মানুহকেইগৰাকীমানেৰেতোনে	manuhkeigɒrakimaneɾetone	মানুহ+কেইগৰাকী+মান+েৰে+তো+নে
50.	মানুহকেইগৰাকীমানেৰেতোচোন	manuhkeigɒrakimaneɾeto∫on	মানুহ+কেইগৰাকী+মান+েৰে+তো+চোন

Table A.1: Some inflectional forms for the noun root মানুহ (manuh *man*)

Assamese Verb Inflection

We found 380 inflectional forms for the Assamese verb root আছ (asɒ : *to be*). Table A.2 lists a few of the obtained inflectional forms. The second column is the inflected word, third column represents the transcribed form in IPA and the fourth column shows the break-up of the word into valid morphological units.

Sl.	Word	IPA	Root+Suffixes
1.	আছোঁ	asõ	আছ+ঁা
2.	আছোঁৱেই	asõwɛi	আছ+ঁা+ৱেই
3.	আছা	asa	আছ+া
4.	আছাই	asai	আছ+াই
5.	আছে	asɛ	আছ+ে
6.	আছেই	asɛi	আছ+েই
7.	আছিলোঁ	asilõ	আছ+িলোঁ
8.	আছিলোঁৱেই	asilõwɛi	আছ+িলোঁ+ৱেই
9.	আছিলোঁহে	asɪlõhɛ	আছ+িলোঁ+হে
10.	আছিলি	asili	আছ+িলি
11.	আছিলিয়েই	asilijɛi	আছ+িলি+য়েই
12.	আছিলা	asila	আছ+িলা
13.	আছিলাই	asilai	আছ+িলা+ই
14.	আছিল	asil	আছ+িল
16.	আছিলোঁ	asilõ	আছ+িলোঁ
17.	আছচোন	asɒson	আছ+চোন
18.	আছোঁচোন	asõson	আছ+ঁা+চোন
19.	আছাচোন	asason	আছ+া+চোন
20.	আছেচোন	asɛson	আছ+ে+চোন
21.	আছিলোঁচোন	asiloson	আছ+িলোঁ+চোন
22.	আছিলিচোন	asilison	আছ+িলি+চোন
23.	আছিলাচোন	asilason	আছ+িলা+চোন
24.	আছিলচোন	asilson	আছ+িল+চোন
25.	আছিলেচোন	asilɛson	আছ+িলে+চোন
26.	আছেতো	asɛṭo	আছ+ে+তো
27.	আছিলোঁতো	asilõṭo	আছ+িলোঁ+তো
28.	আছিলিতো	asiliṭo	আছ+িলি+তো
29.	আছিলাতো	asilaṭo	আছ+িলা+তো
30.	আছিলতো	asilṭo	আছ+িল+তো
31.	আছিলেতো	asilɛṭo	আছ+িলে+তো
32.	আছহঁক	ashŏk	আছ+হঁক
33.	আছনে	asɒnɛ	আছ+নে
34.	আছোঁনে	asõnɛ	আছ+ঁা+নে
35.	আছানে	asanɛ	আছ+া+নে

Table A.2: Some inflectional forms of the root verb আছ (asɒ : *to be*)

Appendix B

Tagset details

B.1 PENN tagset

Sl.	POS tag	Description	Example
1	CC	coordinating conjunction	and
2	CD	cardinal number	1, third
3	DT	determiner	the
4	EX	existential there	*there* is
5	FW	foreign word	d'hoevre
6	IN	preposition/subordinating/conjunction	in, of, like
7	JJ	adjective	green
8	JJR	adjective, comparative	greener
9	JJS	adjective, superlative	greenest
10	LS	list marker	1)
11	MD	modal	could, will
12	NN	noun, singular or mass	table
13	NNS	noun plural	tables
14	NNP	proper noun, singular	John
15	NNPS	proper noun, plural	Vikings
16	PDT	predeterminer	*both* the boys
17	POS	possessive ending	friend*'s*
18	PRP	personal pronoun	I, he, it
19	PRP$	possessive pronoun	my, his
20	RB	adverb	however, usually, here, good
21	RBR	adverb, comparative	better
22	RBS	adverb, superlative	best
23	RP	particle	give *up*
24	TO	to	*to* go, *to* him
25	UH	interjection	uhhuhhuhh
26	VB	verb, base form	take
27	VBD	verb, past tense	took

28	VBG	verb, gerund/present participle	taking
29	VBN	verb, past participle	taken
30	VBP	verb, sing. present, non-3d	take
31	VBZ	verb, 3rd person sing. present	takes
32	WDT	wh-determiner	which
33	WP	wh-pronoun	who, what
34	WP$	possessive wh-pronoun	whose
35	WRB	wh-abverb	where, when

Table B.1: Penn tagset

B.2 BNC tagset

AJ0	Adjective (general or positive) (e.g. *good, old, beautiful*)
AJC	Comparative adjective (e.g. *better, older*)
AJS	Superlative adjective (e.g. *best, oldest*)
AT0	Article (e.g. *the, a, an, no*)
AV0	General adverb: an adverb not subclassified as AVP or AVQ (see below) (e.g. *often, well, longer* (adv.), *furthest.*
AVP	Adverb particle (e.g. *up, off, out*)
AVQ	Wh-adverb (e.g. *when, where, how, why, wherever*)
CJC	Coordinating conjunction (e.g. *and, or, but*)
CJS	Subordinating conjunction (e.g. *although, when*)
CJT	The subordinating conjunction *that*
CRD	Cardinal number (e.g. *one, 3, fifty-five, 3609*)
DPS	Possessive determiner-pronoun (e.g. *your, their, his*)
DT0	General determiner-pronoun: i.e. a determiner-pronoun which is not a DTQ or an AT0.
DTQ	Wh-determiner-pronoun (e.g. *which, what, whose, whichever*)
EX0	Existential there, i.e. *there* occurring in the *there is* ... or *there are* ... construction
ITJ	Interjection or other isolate (e.g. *oh, yes, mhm, wow*)
NN0	Common noun, neutral for number (e.g. *aircraft, data, committee*)
NN1	Singular common noun (e.g. *pencil, goose, time, revelation*)
NN2	Plural common noun (e.g. *pencils, geese, times, revelations*)
NP0	Proper noun (e.g. *London, Michael, Mars, IBM*)
ORD	Ordinal numeral (e.g. *first, sixth, 77th, last*) .
PNI	Indefinite pronoun (e.g. *none, everything, one* [as pronoun], *nobody*)
PNP	Personal pronoun (e.g. *I, you, them, ours*)
PNQ	Wh-pronoun (e.g. *who, whoever, whom*)
PNX	Reflexive pronoun (e.g. *myself, yourself, itself, ourselves*)
POS	The possessive or genitive marker *'s* or *'*
PRF	The preposition *of*
PRP	Preposition (except for *of*) (e.g. *about, at, in, on, on behalf of, with*)
PUL	Punctuation: left bracket - i.e. *(* or *[*
PUN	Punctuation: general separating mark - i.e. *. , ! , : ;* - or *?*
PUQ	Punctuation: quotation mark - i.e. *'* or *"*
PUR	Punctuation: right bracket - i.e. *)* or *]*
TO0	Infinitive marker *to*
UNC	Unclassified items which are not appropriately considered as items of the English lexicon.
VBB	The present tense forms of the verb BE, except for *is, 's*: i.e. *am, are, 'm, 're* and *be* [subjunctive or imperative]
VBD	The past tense forms of the verb BE: *was* and *were*
VBG	The -ing form of the verb BE: *being*
VBI	The infinitive form of the verb BE: *be*
VBN	The past participle form of the verb BE: *been*
VBZ	The -s form of the verb BE: *is, 's*
VDB	The finite base form of the verb BE: *do*
VDD	The past tense form of the verb DO: *did*

VDG	The -ing form of the verb DO: *doing*
VDI	The infinitive form of the verb DO: *do*
VDN	The past participle form of the verb DO: *done*
VDZ	The -s form of the verb DO: *does*, *'s*
VHB	The finite base form of the verb HAVE: *have*, *'ve*
VHD	The past tense form of the verb HAVE: *had*, *'d*
VHG	The -ing form of the verb HAVE: *having*
VHI	The infinitive form of the verb HAVE: *have*
VHN	The past participle form of the verb HAVE: *had*
VHZ	The -s form of the verb HAVE: *has*, *'s*
VM0	Modal auxiliary verb (e.g. *will*, *would*, *can*, *could*, *'ll*, *'d*)
VVB	The finite base form of lexical verbs (e.g. *forget*, *send*, *live*, *return*) [Including the imperative and present subjunctive]
VVD	The past tense form of lexical verbs (e.g. *forgot*, *sent*, *lived*, *returned*)
VVG	The -ing form of lexical verbs (e.g. *forgetting*, *sending*, *living*, *returning*)
VVI	The infinitive form of lexical verbs (e.g. *forget*, *send*, *live*, *return*)
VVN	The past participle form of lexical verbs (e.g. *forgotten*, *sent*, *lived*, *returned*)
VVZ	The -s form of lexical verbs (e.g. *forgets*, *sends*, *lives*, *returns*)
XX0	The negative particle *not* or *n't*
ZZ0	Alphabetical symbols (e.g. *A*, *a*, *B*, *b*, *c*, *d*)

Table B.2: BNC basic tagset

B.3 Xobdo tagset

Sl.	Major POS	Minor POS
1	Noun	Common Noun
2		Proper Noun
3		Material Noun
4		Verbal Noun
5		Abstract Noun
6	Pronoun	-
7	Adjective	Proper Adjective
8		Verbal Adjective
9		Adjective of Adjective
10		Adverb
11	Verb	Transitive Verb
12		Intransitive Verb
13	Others	Ad-position
14		Interjection
15		Conjunction

Table B.3: Xobdo tagset

B.4 TUTaget–F

Assamese Part of Speech tagset *Flat*
Developed by: Dhruba Jyoti Das and Utpal Sharma
Department of Computer Science and Engineering
Tezpur University, Napaam
INDIA - 784028
Email: utpal@tezu.ernet.in

Sl.	Assamese name		English	Tag	Example
1.	ekbasan bisheshya	jatibasok	singular common noun	CN0	*manuh*
2.	ekbasan bisheshya	jatibasok	nominative singular common noun	CN1	*manuhe*
3.	ekbasan bisheshya	jatibasok	accusative singular common noun	CN2	*manuhak*
4.	ekbasan bisheshya	jatibasok	instrumental singular common noun	CN3	*manuhere*
5.	ekbasan bisheshya	jatibasok	dative singular common noun	CN4	*manuhala'i*
6.	ekbasan bisheshya	jatibasok	genitive singular common noun	CN5	*manuhar*
7.	ekbasan bisheshya	jatibasok	locative singular common noun	CN6	*manuhat*
8.	bahubasan bisheshya	jatibasok	plural common noun	CNS0	*manuhbor*
9.	bahubasan bisheshya	jatibasok	nominative plural common noun	CNS1	*manuhbore*
10.	bahubasan bisheshya	jatibasok	accusative plural common noun	CNS2	*manuhborak*
11.	bahubasan bisheshya	jatibasok	instrumental plural common noun	CNS3	*manuhborere*
12.	bahubasan bisheshya	jatibasok	dative plural common noun	CNS4	*manuhborala'i*
13.	bahubasan bisheshya	jatibasok	genitive plural common noun	CNS5	*manuhborar*
14.	bahubasan bisheshya	jatibasok	locative plural common noun	CNS6	*manuhborat*
15.	ekbasan bisheshya	sangyabasok	singular proper noun	PN0	*ram*
16.	ekbasan bisheshya	sangyabasok	nominative singular proper noun	PN1	*rame*
17.	ekbasan bisheshya	sangyabasok	accusative singular proper noun	PN2	*ramak*
18.	ekbasan bisheshya	sangyabasok	instrumental singular proper	PN3	*ramere*
19.	ekbasan bisheshya	sangyabasok	dative singular proper noun	PN4	*ramala'i*

20.	ekbasan sangyabasok bisheshya	genitive singular proper noun	PN5	*ramar*
21.	ekbasan sangyabasok bisheshya	locative singular proper noun	PN6	*brindabanat*
22.	bastubasok bisheshya bisheshya	singular material noun	MN0	*saul*
23.	bastubasok bisheshya bisheshya	nominative singular material noun	MN1	*dhane*
24.	bastubasok bisheshya bisheshya	accusative singular material noun	MN2	*bayuk*
25.	bastubasok bisheshya bisheshya	instrumental singular material noun	MN3	*saulere*
26.	bastubasok bisheshya bisheshya	dative singular material noun	MN4	*dhanala'i*
27.	bastubasok bisheshya	genitive singular material noun	MN5	*saular*
28.	bastubasok bisheshya bisheshya	locative singular material noun	MN6	*saulat*
29.	bastubasok bisheshya bisheshya	plural material noun	MNS0	*dhanbor*
30.	bahubasan bastubasok bisheshya	nominative plural material noun	MNS1	*dhanbore*
31.	bastubasok bisheshya	accusative plural material noun	MNS2	*saulborak*
32.	bahubasan bastubasok bisheshya	instrumental plural material noun	MNS3	*tamolborere*
33.	bahubasan bastubasok bisheshya	dative plural material noun	MNS4	*dhanborala'i*
34.	bahubasan bastubasok bisheshya	genitive plural material noun	MNS5	*saulborar*
35.	bahubasan bastubasok bisheshya	locative plural material noun	MNS6	*saulborat*
36.	gunobasok bisheshya	abstract noun	AN0	*khyama, bishad*
37.	gunobasok bisheshya	nominative abstract noun	AN1	*sadhutai, ekotai*
38.	gunobasok bisheshya	accusative abstract noun	AN2	*sadhutak, birattak*
39.	gunobasok bisheshya	instrumental abstract noun	AN3	*sadhutare, birattare*
40.	gunobasok bisheshya	dative abstract noun	AN4	*khyamala'i*
41.	gunobasok bisheshya	genitive abstract noun	AN5	*khyamar, sadhutar*
42.	gunobasok bisheshya	locative abstract noun	AN6	*bhadratat*
43.	kriyabasok bisheshya	verbal noun	VN0	*bhraman*
44.	kriyabasok bisheshya	nominative verbal noun	VN1	*gamane*
45.	kriyabasok bisheshya	accusative verbal noun	VN2	*bhramanak*
46.	kriyabasok bisheshya	instrumental verbal noun	VN3	*bhramanere*
47.	kriyabasok bisheshya	dative verbal noun	VN4	*bhramanala'i*
48.	kriyabasok bisheshya	genitive verbal noun	VN5	*bhramanar*
49.	kriyabasok bisheshya	locative verbal noun	VN6	*bhramanat*
50.	kalbasok bisheshya	time indicative noun	TN0	*puwa, gadhuli, second*
51.	kalbasok bisheshya	time indicative noun, nominative	TN1	*puwai, ghantai*
52.	kalbasok bisheshya	time indicative noun, accusative	TN2	*ghantak, basarak*

53. kalbasok bisheshya	time indicative noun, instrumental	TN3	*ghantare*
54. kalbasok bisheshya	time indicative noun, dative	TN4	*sandhiyala'i*
55. kalbasok bisheshya	time indicative noun, genitive	TN5	*sandhiyar, gadhulir*
56. kalbasok bisheshya	time indicative noun, locative	TN6	*secondat, ghantat*
57. kalbasok bisheshya, bahubasan	time indicative noun, plural	TNS0	*dinbor, basarbor*
58. kalbasok bisheshya, bahubasan	time indicative noun, plural, nominative	TNS1	*dinbore*
59. kalbasok bisheshya, bahubasan	time indicative noun, plural, accusative	TNS2	*dinborok*
60. kalbasok bisheshya, bahubasan	time indicative noun, plural, instrumental	TNS3	*dinborere*
61. kalbasok bisheshya, bahubasan	time indicative noun, plural, dative	TNS4	*dinborala'i*
62. kalbasok bisheshya, bahubasan	time indicative noun, plural, genetiv	TNS5	*dinborar*
63. kalbasok bisheshya, bahubasan	time indicative noun, plural, locative	TNS6	*dinkeitat*
64. samastibasok bisheshya, ekbasan	group indicative noun, singular	SN0	*sabha*
65. samastibasok bisheshya, ekbasan	group indicative noun, singular, accusative	SN1	*sabhai*
66. samastibasok bisheshya, ekbasan	group indicative noun, singular, accusative	SN2	*sabhak*
67. samastibasok bisheshya, ekbasan	group indicative noun, singular, instrumental	SN3	*sabhare*
68. samastibasok bisheshya, ekbasan	group indicative noun, singular, dative	SN4	*sabhala'i*
69. samastibasok bisheshya, ekbasan	group indicative noun, singular, genitive	SN5	*sabhar, parishadr*
70. samastibasok bisheshya, ekbasan	group indicative noun, singular, locative	SN6	*sabhat, parishadt*
71. samastibasok bisheshya, bahubasan	group indicative noun, plural	SNS0	*sabhasamuh*
72. samastibasok bisheshya, bahubasan	group indicative noun, plural, nominative	SNS1	*samitibore*
73. samastibasok bisheshya, bahubasan	group indicative noun, plural, accusative	SNS2	*sabhasamuhak*
74 samastibasok bisheshya, bahubasan	group indicative noun, plural, instrumental	SNS3	*kamitisamuhere*
75. samastibasok bisheshya, bahubasan	group indicative noun, plural, dative	SNS4	*sabhaboria'i*
76. samastibasok bisheshya, bahubasan	group indicative noun, plural, genitive	SNS5	*parishadsamuhar*
77. samastibasok bisheshya, bahubasan	group indicative noun, plural, locative	SNS6	*sabhaborat*
78. bisheshyar bisheshan	adjective of noun	NA	*dangor, saru, udanda*
79. bisheshanor bisheshan	adjective of adjective	AA	*bor, ati, kiskisia, tiktikia*

80. kriya bisheshan	adverb	VA	*kharkoi, sonkale*
81. byektibodhak sorbonam, ekbasan	singular personal pronoun	PP0	*moi, si*
82. byektibodhak sorbonam, ekbasan	nominative singular personal pronoun	PP1	*ekhete, tekhete*
83. byektibodhak sorbonam, ekbasan	accusative singular personal pronoun	PP2	*mok, eik*
84. byektibodhak sorbonam, ekbasan	instrumental singular personal pronoun	PP3	*mor dara*
85. byektibodhak sorbonam, ekbasan	dative singular personal pronoun	PP4	*mola'i, tola'i*
86. byektibodhak sorbonam, ekbasan	genitive singular personal pronoun	PP5	*mor, tar*
87. byektibodhak sorbonam, bahubasan	plural personal pronoun	PPS0	*ami, tomalok*
88. byektibodhak sorbonam, bahubasan	nominative plural personal pronoun	PPS1	*teoloke*
89. byektibodhak sorbonam, bahubasan	accusative plural personal pronoun	PPS2	*tomalokok*
90. byektibodhak sorbonam, bahubasan	instrumental plural personal pronoun	PPS3	*amardara*
91. byektibodhak sorbonam, bahubasan	dative plural personal pronoun	PPS4	*amala'i, sihatola'i*
92. byektibodhak sorbonam, bahubasan	genitive plural personal pronoun	PPS5	*amar, tomalokor*
93. kalbodhak sorbonam	pronominal adverb of time	PAT	*ajire, kalilai*
94. bisheshanbodhak sorbonam	pronominal adjective	PA	*jene, tene, jiman, eito*
95. sthanbodhak sorbonam	pronominal adverb of place	PAP	*eyat, tat*
96. jojak abyai	conjunction	CNJ	*aru, karone,*
97. prithokbodhak abyai	disjunctive	DN	*naiba, ba,*
98. bhabbodhak abyai	interjection	IN	*haai, ahaa,*
99. sambodhanbodhak abyai	vocatives	RP	*a, hera, hero*
100.prasnobodhak abyai	interrogative	IP	*ki, kene,*
101.samidhanbodhak abyai	-	AP	*hoi, baru,*
102.anushangik abyai	collateral particle	PPS	*soite, son, dore,*
103.sandehbodhak abyai	-	CF	
104.nitya bortoman	present indefinite	VNP	*moi paro,*
105.swarup bortoman	present continuous	VBP	*moi poriso,*
106.swarup bhut	present perfect	VCP	*moi khalo,*
107.parokhya bhut	past indefinite	VDP	*moi khaisilo*
108.sambhabya bhut kal	conditional past	VFP	*moi khaloheten*
109.bhabishyat kal	future tense	VGP	*moi kham*
110.nimitarthok sangya	gerund (class1)	VHP	*apel khaboloi bor sowad.*
111.nimitarthok sangya	gerund (class2)	VHP1	*bera diote hatkhon katile.*
112.owsityya gyapok kriya, bortoman	-	VJP	*lage, usit,*

113.owsityya gyapok kriya, atit	-	VJD	*lagisil*
114.owsityya gyapok kriya, bhabishyat	-	VJF	*lagibo*
115.asamapika kriya	non finite form of verb (class1)	VNFI	*khai, goi, hoi,*
116.asamapika kriya	non finite form of verb (class2)	VNFA	*moi khowa nai.*
117.asamapika kriya	non finite form of verb (class3)	VNFP	*mor khowa nohol.*
118.aniyomit kriya	irregular verb, present indefinite	VYP	*thake*
119.aniyomit kriya	irregular verb, present continuous	VXP	*ase*
120.aniyomit kriya	irregular verb, past continuous	VOP	*asil, thakisil*
121.aniyomit kriya	irregular verb, future continuous	VPP	*thakibo*
122.pasoni kriya, nitya bortoman	causative verb, present indefinite	VQP	*karaow, karowa*
123.pasoni kriya, swarup bortoman	causative verb, present continuous	VRP	*karaiso, karaisa*
124.pasoni kriya, atit, parokhya	causative verb, past Indefinite	VRD	*karaisilo,*
125.pasoni kriya, atit, swarup	causative verb, present perfect	VQD	*karalo, karala*
126.pasoni kriya, atit, sambhabya	causative verb, conditional past	VOD	*karaloheten,*
127.pasoni kriya, bhabishyat	causative verb, future indefinite	VQF	*karam, karaba*
128.ganitik sinho	mathematical symbol	SYM	/,*,−,+,>,<,
129.sankhya	cardinal number	CNU	*1, 2, 3, ek, dui, tini*
130.sankhya	ordinal number	ONU	*prothom, ditiyo, tritiyo*
131.	dash	PUD	−−
132.	opening bracket, left	PUL	(
133.	closing bracket, right	PUR	)
134.	single quotation, opening	PUQO	‘
135.	single quotation, closing	PUQC	’
136.	double quotation, opening	PUQO	“
137.	double quotation, closing	PUQC	”
138.	fullstops, exclamation, question mark	PUN	|,!,?
139.	comma	PUK	,
140.kridanta karta	derived Subject	VPO	*thakota, shuota, karota.*
141.	deverbal noun	DVN	*Khowato, thokato, howato*
142.	deverbal noun, genitive	DVN-GN	*khowar, shuwar, porar, khowar, jonowar.*
143.	deverbal noun, locative	DVN-LOC	*mor khowat apotti nai*
144.	deverbal noun, accusative	DVN-ACC	*akol khowak loi eman hulosthulkhon nokoriba-son*

145.	passiviser	PAS	*dekha jai, kora hol*
146.	negativiser	NES	*dekha najai, kora nohol*
147.	finite verb, present (Passive)	FVP0	*ear pora sobdoto suni (suna jai)*
148.	finite verb, past (passive)	FVP1	*ear pora ghorto dekhisil (dekha goisil)*
149.	finite verb, future (passive)	FVP2	*ear pora ghorto dekhibo (dekha jabo)*
150.	deverbal adjective	DVA	*kuwar pora khowa pani tuli asilo.*
151.	adjective Comparative	AD-COM	*rita gitatkoi dhunia.*
152.	adjective, Superlative	AD-SUP	*rita sokolotkoi dhunia.*
153.	be-possesive, present	VBP0	*mor eta kolom ase.*
154.	be-possesive, past	VBP1	*mor eta dhunia sola asil.*
155.	be-possesive, future	VBP2	*ebosor pasot mor eta dhunia ghor thakibo.*
156.	verb, non finite+gerund	FPT	*nokorakoi, nojowakoi, noporakoi, howakoi.*
157.	verb, nonfinite+infinitival	FQT	*koribologia, subologia, jabologia, poribologia.*
158.	verbal agreement	VNZ	*porhok, suwak, khaok, jaok.*
159.	verbal complex	VLP	*khale, porhile, gole, sule.*
160.	non-finite verb	VKP	*najaotei, nohowtei, nakhaotei.*
161.	collective and fractional word	CFW	*adha, der, caretini, satkara, ezor, ehal.*
162.	year	YER	*1902, 1941, 1973,1982.*
163.	noun (unit of measurement)	N-UME	*gram, kilogram, ton, liter, bigha, kotha.*
164.	negation used after non finite form of verb	NAI	*dekha nai, khowa nai.*
165.kridonto bishesan	derived adjective	DA	*aha mahot, jowa mahot, aha sombar, porisit manuh, anusthit sabha.*
166.	unit of measurement of rupees	UMR	*caa, hazar, lakh.*
167.sankhyabodhak sobdo	cardinals	CAR	*eta, duta, ebar, dubar, edin, dudin.*
168.	case marker (nominative)	NCM	*i, e.*
169.	case marker (accusative)	ACM	*k.*
170.	case marker (dative)	DCM	*al'i.*
171.	case marker (locative)	LCM	*t.*
172.	case marker (genetic)	GCM	*r.*

Table B.4: TUTaget–F

B.5 LDC-IL tagset

Version 0.3, 3rd Oct 2009

Type	Attributes
Noun(N)	
Common(NC)	Gender, Number, Case, Case Marker, Emphatic, Definiteness, Inclusive, Exclusive, Topic, Confirmative, Honorificity
Proper(NP)	Number, Case, Case Marker, Emphatic, Definiteness, Inclusive, Exclusive, Topic, Confirmative, Honorificity
Verbal(NV)	Gender, Number, Case, Case Marker, Emphatic, Definiteness, Inclusive, Exclusive, Topic, Confirmative, Honorificity, Negative
Spatio-temporal (NST)	Number, Case, Case Marker, Emphatic, Definiteness, Inclusive, Exclusive, Topic, Confirmative
Pronoun(P)	
Pronominal (PPR)	Gender, Number, Person, Case, Case marker, Emphatic, Definiteness, Inclusive, Exclusive, Topic, Confirmative, Dimension, Honorificity
Reflexive (PRF)	Gender, Number, Case, Case marker, Emphatic, Definiteness, Inclusive, Exclusive, Topic
Reciprocal (PRC)	Number, Case, Case marker, Emphatic, Definiteness, Dimension, Honorificity
Relative (PRL)	Gender, Number, Case, Case marker, Emphatic, Honorificity, Definiteness
Wh-pronoun (PWH)	Gender, Number, Case, Case marker, Topic, Emphatic, Honorificity, Definiteness
Demonstrative (D)	
Absolutive (DAB)	Gender, Number, Case, Case Marker, Dimension, Emphatic, Definiteness, Inclusive, Exclusive, Topic, Confirmative
Relative Demonstrative (DRL)	Gender, Number, Case, Case Marker, Emphatic, Definiteness
Wh-demostrative (DWH)	Gender, Number, Case, Case Marker, Emphatic, Definiteness
Nominal Modifier (J)	
Adjective (JJ)	Gender, Number, Emphatic, Confirmative
Quantifier (JQ)	Number, Emphatic, Definiteness, Numeral
Intensifier (JINT)	
Verb (V)	
Main Verb (VM)	Person, Tense, Aspect, Mood, Finiteness, Inclusive, Exclusive, Topic, Confirmative, Honorificity, Negative, Deictic, Emphatic
Auxiliary Verb (VA)	Person, Tense, Aspect, Mood, Finiteness, Inclusive,

	Exclusive, Topic, Confirmative, Honorificity, Negative, Deictic, Emphatic
Adverb(A)	
Manner (AMN)	Topic, Emphatic
Location (ALC)	Case, Case Marker, Topic, Confirmative, Inclusive, Exclusive, Emphatic
Post- position(PP)	
Post- position(PP)	Emphatic, Inclusive , Exclusive
Particle (C)	
Co-ordinating (CCD)	
Subordinating (CSB)	Emphatic
Interjection (CIN)	
(Dis)Agreement (AGR)	
Delimitative (DLIM)	
Dedative (DED)	
Dubitative (DUB)	
Similative (SIM)	Emphatic
Others (CX)	Emphatic
Numeral (NUM)	
Cardinal (CRD)	Case, Case- marker, Definiteness
Ordinal (ORD)	
Date (DT)	
Unit (UNT)	
Residual(RD)	
Foreign Word (RDF)	
Symbol (RDS)	
Reduplication (RDP)	
Unknown (UNK)	
Punctuation (PU)	

Table B.5: LDC-IL tagset

Attribute	Values
Person(*PER*)	First(1) Second(2) Third(3)
Number(*NUM*)	Singular(*sg*) Plural(*pl*)
Gender(*GEN*)	Masculine(*mas*) Feminine(*fem*)
Case(*CS*)	Direct(*dir*) Oblique(*obl*)
Case Marker(*CSM*)	Nominative(*nom*) Accusative(*acc*) Dative(*dat*) Instrumental(*ins*)
	Ablative(*abl*) Genitive(*gen*) Locative(*loc*) Vocative(*voc*)
	Terminative(*term*) Allative(*alla*) Comparative(*comp*) Vialis(*via*)
Tense(*TNS*)	Present(*prs*) Past(*pst*) Future(*fut*)
Aspect(*ASP*)	Imperfective(*ipfv*) Perfect(*prf*)
Mood(*MOOD*)	Declarative(*dcl*) Subjunctive(*sbj*) Conditional(*cnd*)
	Imperative(*imp*) Presumptive(*psm*) Abilitative(*abl*)
	Habitual(*hab*) Interrogative(*int*) Obligatory(*Obl*)
Optative(*OPT*)	Finiteness(*FIN*) Finite(*fin*) Infinite(*inf*) Non –finite(*nfn*)
Dimension(*DIM*)	Proximal(*prx*) Distal(*dst*) Remote(*rem*)

Table B.6: Attribute and their values in LDC-IL Tagset.

B.6 BIS tagset

Linguistic Standards for Indian Language (IL) Corpora Annotation
(June 12, 2010)

Top level	Sub level 1	Label	Annotation Convention	Examples
Noun		N	N	lorA, rojA, kitaaba
	Common	NN	N_NN	kitaaba, kalama, bohii
	Proper	NNP	N_NNP	Mohan, ravi, rashmi
	Nloc	NST	N_NST	Uupara, tol, aag, pis
Pronoun		PR	PR	iyat, tAt, jun
	Personal	PRP	PR_PRP	moi, tumi, teu, amar
	Reflexive	PRF	PR_PRF	nijei, swayam,
	Relative	PRL	PR_PRL	Jun, jak, jetiA, jote
	Reciprocal	PRC	PR_PRC	Itue-xituk
	Wh-word	PRQ	PR_PRQ	Kuna, ketiya, kote
	Indefinite	PRI	PR_PRI	Kunuba, kiba
Demonstrative		DM	DM	ji, jun, iyat,
	Deictic	DMD	DM_DMD	taat, iyat
	Relative	DMR	DM_DMR	jun, jaar
	Wh-word	DMQ	DM_DMQ	kaar, kun
	Indefinite	DMI	DM_DMI	Kunuba, kiba
Verb	Main	VM	V_VM	Karibalai, rakhAi, karichE
		VF	V_VM_VF	karichE, dilE, dharichE
		VNF	V_VM_VNF	rakhAi, Ahi, sAni
		VINF	V_VM_VINF	Kariba, Ahiba, tuliba
		V?		paThOwA, karATO, gaDh'Ara, karATOhE kOwAta
	Auxiliary	VAUXF	V_VAUX_VAUXF	Achila, ga'la, AchE, thAkE
Adjective		JJ	JJ	dhuniya, bhAl, dangor
Adverb		RB	RB	khorkoi, xunkale
Postposition		PSP	PSP	logot, porA, dArA
Conjunction		CC	CC	aru, nohole, kintu
	Co-ordinator	CCD	CC_CCD	Jodi, karone, tente, je
	Subordinator	CCS	CC_CCS	Jodi, karone, tente, je
Particles		RP	RP	sun, dekhun, biraat
	Default	RPD	RP_RPD	sun, dekhun, bAru
	Interjection	INJ	RP_INJ	Are, oh, o
	Intensifier	INTF	RP_INTF	birAt, khuub
	Negation	NEG	RP_NEG	nahai, na, bihane
Quantifiers		QT	QT	olop, bahuta, kisu, eka, pratham

	General	QTF	QT_QTF	olop, bahuta, kisu
	Cardinals	QTC	QT_QTC	eka, dui, tiini
	Ordinals	QTO	QT_QTO	pratham, dritiya
Residuals		RD	RD	
	Foreign word	RDF	RD_RDF	
	Symbol	SYM	RD_SYM	$, &, *, ,
	Punctuation	PUNC	RD_PUNC	., : ;
	Unknown	UNK	RD_UNK	
	Echowords	ECH	RD_ECH	(Paanii-saanii, bhaat- saat)

Table B 7: BIS tagset

B.7 AnnCora tagset

Sl.	Category	Types	Attributes
1	Noun(N)	Common(C)	1,2,5,6,11,12,13,14
		Proper(P)	1,2,5,6,11,12,13,14
		Verbal(V)	5,6,11,12
		Spatiotemporal(ST)	5,6,11,12,13
2	Verb(V)	Main(M)	1,2,3,4,7,8,9,12,13,16
		Auxiliary(A)	1,2,3,4,7,8,9,12,13,16
3	Pronoun(P)	Pronominal (PR)	1,2,3,5,6,10,11,12,13,15,16
		Reflexive(RF)	1,5,6,11,12,13,15,16
		Reciprocal(RC)	1,5,6,11,12
		Relative (RL)	2,5,6,10,11,12,13,15,16
		Wh (Wh)	2,5,6,10,11,12,13,15,16
4	Nominal Modifiers(J)	Adjective(J)	1,2,5,13
		Quantifiers(Q)	1,2,5,11,12,13,17
5	Demonstrative(D)	Absolute (AB)	2,5,12,14
		Relative (RL)	2,5,12,14
		Wh (WH)	2,5,12,14
6	Adverb(A)	Manner (MN)	5,6,11,12,13
		Location (LC)	5,6,11,12,13
7	Participle(L)	Adjectival (RL)	1,2,4,12,13
		Adverbial (V)	12,13
		Nominal (N)	1,2,4,7,12,13
		Conditional (C)	12,13,18
8	Post-positions(PP)		1,2,6,11,12
9	Paticles(C)	Coordinating	
		Coordinating (CD)	
		Subordinating (SB)	
		Classifier (CL)	12
		Interjection (IN)	
		Others (X)	12
10	Punctuation (PU)		
11	Residual (RD)	Foreign word(F)	
		Symbols (S)	
		other (X)	

Table B.8: AnnCora tagset

Sl.	Arttibute	Value
1	Gender	Masculine (mas), Feminine (fem),Neuter (neu)
2	Number (Num)	Singular (sg), Plural (pl), Dual (du)
3	Person (Per)	First (1), Second (2), Third (3)
4	Tense (Tns)	Present (prs), Past (pst), Future (fut)
5	Case (Cs)	Direct (dir), Oblique (obl)
6	Case-marker (Csm)	Ergative (erg), Accusative (acc), Instrumental (ins),

		Dative (dat), Genitive (gen), Sociative (soc), Locative (loc), Ablative (abl), Benefective (bnf), Vocative (voc), Purposive (pur)
7	Aspect (Asp)	Simple (sim), Progressive (prg), Purposive (prf)
8	Mood (Mood)	Declarative (dcl), Subjunctive (sbj), Conditional (cnd), Imperative (imp), Presumptive (psm), Abilitative (abt), Habitual (hab)
9	Finiteness (Fin)	Finite (fin), Non-finite (nfn), Infinite (inf)
10	Distributive (Dstb)	Yes (y), No (n)
11	Emphatic (Emph)	Yes (y), No (n)
12	Negative (Neg)	Yes (y), No (n)
13	Distance (Dist)	Proximal (prx), Distal(dst), Sequel (seq)
14	Incl/Excl (Set)	Inclusive (inl), Exclusive (exl)
15	Honorificity (Hon)	Yes (y), No (n)
16	Numeral (Nml)	Ordinal (ord), Cardinal (crd), Non-numeral (nnm)
17	Realis(Rls)	Yes (y), No (n)

Table B.9: Attribute and their values

Appendix C

Rules of link grammar for Assamese

The grammar spread over four files: affix information file, constituent knowledge file, dictionary file and the post-processing file. The main file, where the rules are stored is called *dictionary* and each of the linking requirements of words is expressed in terms of connectors in the dictionary file. The linking requirements for each word consist of connector names followed by directions of the links, parentheses to denote precedence and & and OR operators that handle more than one outgoing or incoming links from the word (if any) and macros. A macro is a kind of user defined variable, where an user can encapsulate collection of frequently used rules for further use. The following are rules developed for Assamese.

```
% Noun rule-set as a dictionary entry
%
% Explanation:
%
% The nouns are open category. The noun and words that are directly related
% to the noun are known as nominal group or noun phrase. (Though some
% researcher deny to use "phrase" as a synonym of "group". According to them "a
% group is an expansion of a word and a phrase is a contraction of a clause". In
% this report, we are not very strict about the use of these two words and based
% on the structure of the considered languages, we belief, the group is an
% expansion of a word.)

% For example : ৰঙা আঁচ থকা কামিজটো লেতেৰা।
%                (rɔŋa ãs tʰɔka kamizto letɛra : The shirt with red strip is dirty.)

% In the sentence, the first three words are the extension to the noun
% word কামিজটো (kamizto : the shirt). The adjective ৰঙা (rɔŋa : red) is
% modifying the noun আঁচ (ãs : strip) and form a group. With the help of
% a particle থকা (tʰɔka), this group indicates specific property to the
% noun কামিজটো (kamizto : the shirt) and form a bigger group. Thus the
```

Figure C.1: Dependency graph for the sentence ৰঙা আঁচ থকা কামিজটো লেতেৰা (rɔŋa ãs tʰɔka kamizto letɛra)

```
% first four words in the sentence forms a nominal group. The last
% word লেতেৰা (letɛra : dirty) is again an adjective but not a member
% of the previous group, instead it marks a stress to the verb which
% is absent in the sentence. The individual PoS tags of the sentence
% is given below.
%              rɔŋa_(Adj) ãs_(NN) tʰɔka_(PAR) kamizto_(NN+Def) letɛra_(Adj)
% The following rules are used to indicate the relation between
% a noun/nominal group with other part of a sentence.


<noun-nn>: N+;
<noun-np>: NP+;
<noun-np-obj>: NPx+;
<noun-pl>: NPl+;
<noun-acc>: [On+];
<noun-pl-indef>: NPli+;
<psp-sub>: PA+;
<psp-obj>: PO-;
<noun-df>: ND-;

<ns-1>: {@NP+} & {R+ & Bs+ & {[[@NP+]]}} & {@NPx+};
<ns-2>: {NOM+} & {@NP+} & {R+ & Bs+ & {[[@NP+]]}};

/words/words.n:
{G-} & {[MG+]}&
(({DG- or [[GN-]]  or [[{@NoM-} & {NoM-}]]} &
(({@MX+} & (JG- or <ns-1>)) or (YS+ or YP+))) or
 AN+ or G+);

/words/words.n.em:
({@NoM-} & {[[@NoM-]]} &
(({Dmc-} & (<ns-2> or Bpm+)) or
(YP+ & {NoM-})  or
(GN+ & or [[NoM+]]);

/words/words.n.np:
({@NoM-} & {@NoM- & {[[@NoM-]]}} &
(({Dmc-} & <ns-1>) or
```

```
(GN+ & (Dm- or [()])) or Up-));

/words/words.n.pl:
({@NoM-} &
{@NoM- &
(({Dmc-}  or (YS+ & {Dm-})) or
(GN+ & (Dm- or [()])) or Up-));


/words/words.n.dm:
/words/words.n.pl:
/words/words.n.np:
/words/words.n.em:
({<noun-np>} &
{<noun-pl>} &
{<noun-pl-indef>} &
{([[{@NoM-} & {NoM-}]] or <ns-1>)});


% Pronoun rule-set as a dictionary entry
%
% Explanation:
%
% Pronouns are words that are substitute for nouns or nominal groups.
% As pronouns are used to refer nouns, they belong to nominal group.


% For example: মই আৰু তুমি তালৈ যাম।
%              (moi aru tumi talɔi zam.  :  I and you will go there.)
%               moi_(PN) aru_(PAR) tumi_(PN) talɔi_(RB) zam_(VB)
```

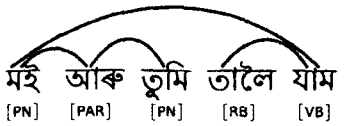


Figure C.2: Dependency graph for the sentence মই আৰু তুমি তালৈ যাম। (moi aru tumi talɔi zam.)

```
% The first three words form a nominal group with the help of a conjunctive
% particle and work as a subject. The subject and the verb  (the last word)
% form a relation. The verb is inflected with first person, that is it caries
% the tense and person information of the head of the subject (the first word
% of the sentence). This two groups (noun and pronoun) can tackle all
% nominal modifiers (NoM).
```

```
<pn-nm-1>: Spn1+ ;
<pn-nm-2>: Spn2+ ;
<pn-nm-3s>: Spn3s+ ;
<pn-nm-3p>: Spn3p+ ;
<pn-gn>:
<pn-dt>·
<pn-nom>:
<pn-istr>:
<pn-loc>:
<pn-dtm>:
({<noun-nn>} & {([[{@NoM-} & {NoM-}]] or <ns-1>)});


% Adverb rule-set as a dictionary entry
%
% Explanation:
%
% From the positional point of view, the Assamese adverb is always placed
% before the verb. However, sometimes the object of the verb may be placed
% between verb and adverb. Adverb is a word that modifies something other
% than a noun. For example, in Figure C.2, the word "taloi" (the fourth word
% from the beginning) is an adverb, that precedes the verb "zam".

/words/words.rb:
({NoM} or NN+)
& (<verb-prefix> or NN+))
or (NP+ & {NoM-})
or (NUM+ & [[NoM+]]);


% Number rule-set as a dictionary entry
%
% Explanation:
%
% Numbers are countable form of any entity. Sometimes numbers are inflected
% with noun suffix.


% For example: একৰ পিছত দুই আহে। (ɛkɔr pisɔt dui ahe : two comes after one.)
%              ɛkɔr_(NUM) pisɔt_(RB) dui_(NUM) ahe_(VB)
```

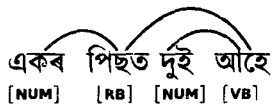


Figure C.3: Dependency graph for the sentence একৰ পিছত দুই আহে। (ɛkɔr pisɔt dui ahe)

```
% In the example, the first word is inflected with nominal suffix and
% the third word which is also a number is in root form.

/words/words.num :
({NoM} & (<ns-2> or NN+))
or (NP+ & {NoM-})
or (NUM+ & or [[NoM+]]);

m.ordm om.ordm: ORDM-;

% Particle rule-set as a dictionary entry
%
% Explanation:
%
% Particles are words that are never inflected. Assamese has a large number
% of particles. Depending on semantics, an Assamese particle can be classified
% into eight different groups. Conjunctive particle can form two or more simple
% sentences or two different clause or group of words. Thus conjunctive particle
% maintains a relation with the mentioned forms. For example, in Figure C.2 the
% second word is a conjunctive particle that maintains relation with first and
% third words.

/words/words.par : PAR+;
/words/words.nom :
({NoM} & (<ns-2> or NN+))
or (NP+ & {NoM-})
or (NUM+ & [[NoM+]]);

% Question word rule-set as a dictionary entry
%
% Explanation:
%
% Question words are inflected with case markers and plural markers or a
% sequence of both. These words are used for interrogation, though in
% some situation it indicates neutrality.

/words/words.qh : QH+;
/words/words.rdp : RDP+;
({NoM} & (<ns-2> or NN+))
or (NUM+ & [[NoM+]]);
```

```
% Verb rule-set as a dictionary entry
%
% Explanation:
%
% Verbs are modified with tense, aspect and mood. Assamese uses three verbs
% to serve the purposes of verb 'be' in English. No form of verb 'be' is used as
% copula in affirmative sentences in the present tense. However, it is present
% in future tense, past tense and in negative constructions in present tense.
% Table 4.5 shows the use of the 'be' verb. Thus it maintains relation
% with its subject and other sub-form of verbs. For example, in Figure C.2
% and C.3, the last words are verb. A verbal group is a sequence of verb and
% its sub-forms that may contain adverb, particle, noun, noun modifier or a
% nominal group.

<verb-prefix>:
(({VMdur-} & {VMneg-}));

<verb-right-side>:
(VMP+ or CCF+ or (VMT+ & VMPP+)
or [()] or [RW+]);

<verb-right-side-comp>:
(VMP+ or CCF+ or (SUB+ or C+)
or VMPP+ or [()] or [RW+]);

<verb-left-side>:
{( O- or CCOB-)}
& {O-} & {@RB-} & {PP-}
& {@RB-} & {[Wi-]};

<verb-left-side-intrans>:
{( P- or CCAJP-)}
& {@RB-} & {PP-} & {@RB-}
& [{(Spn3s- or Sn-)}]
& {[Wi- or C- or CC-]};

/words/words.v.intransitive:
(<verb-prefix> & {@RB-}
& <verb-left-side-intrans>
& {VMT+}
& ( <verb-right-side> or <verb-right-side-comp> ))
or
(VFUT- & {@RB-}
& <verb-left-side-intrans>
& VMT+
& {(CCF+  or  VMPP+  or  RW+ )}):
```

```
% Compound word rule-set as a dictionary entry
%
% Explanation:
%
% Compound words are words consisting of more than one root. based on word
% formation process, they are of three kinds - closed (Example: Greenhouse),
% hyphenated (Example: Green-house) and free form (Example: Green house).
% The first two types do not cause difficulty, as they are only one word
% long and are supposed to be handled properly. The third form may have
% minimum two links - one link for its sub-words and other links to the
% dependent.

/words/words.v.compound:
<verb-prefix>
& {@RB-}
& K-
& <verb-left-side>
& {VMT+}
& <verb-right-side>;

<nom-par-pred>: (CCAPR- or CCAP1+);

<nn-comp>: ({EA- or EF+} & (({[[@Ec-]]} & {Xc+} & Ah+) or
(Pa- & {@MV+}))) or [[AN+]]
or [[({@AN-} & {@A-} & (({D-} & <ns-1> & (<ns-2> or B*m+)) or
 U-)) or ((YS+ or YP+) & {@AN-} & {@A-} & {D-})]];
```

~~*~~

Bibliography

[1] Zhang, T. *et al.* Text chunking based on a generalization of winnow. *The Journal of Machine Learning Research* **2**, 615–637, 2002.

[2] Kudo, T. & Matsumoto, Y. Chunking with support vector machines. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies.* 1–8, Pittsburgh, Pennsylvania, 2001.

[3] Sleator, D. & Temperley, D. Parsing English with a link grammar. Tech. Rep., Department of Computer Science, Carnegie Mellon University, 1991.

[4] Nivre, J. & Hall, J. Maltparser : A language-independent system for data driven dependency parsing. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories.* 137–148, Barcelona, 2005.

[5] McDonald, R. *et al.* Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing.* 523–530, Vancouver, Canada, 2005.

[6] Covington, M. A. A dependency parser for variable-word-order languages. Tech. Rep., The University of Georgia, 1990.

[7] Jurafsky, D. & Martin, J. H. *Speech and Language Processing, An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition,* Pearson Education, 2000, 2nd edn.

[8] Sharma, U. *et al.* Unsupervised learning of morphology for building lexicon for a highly inflectional language. In *Proceedings of the ACL Workshop on Morphological and Phonological Learning.* 1–6, Philadelphia, USA, 2002.

[9] Das, M. *et al.* Design and implementation of a spell checker for Assamese. In *Proceedings of Language Engineering Conference.* 156–162, Hyderabad, India, 2002.

[10] Sharma, U. *et al.* Root word stemming by multiple evidence from corpus. In *Proceedings of 6th International Conference on Computational Intelligence and Natural Computing.* 1593–1596, North Carolina, USA, 2003.

[11] Sharma, U. *Unsupervised Learning of Morphology of A Highly Inflectional Language.* Ph.D. thesis, Tezpur University, 2007.

[12] Sharma, U. *et al.* Acquisition of morphology of an Indic language from text corpus. *ACM Transactions of Asian Language Information Processing* **7** (3), 1–33, 2008.

[13] Crystal, D. *The Cambridge encyclopedia of the English language*, Ernst Klett Sprachen, 2004.

[14] Biber, D. Representativeness in corpus design. *Literary and linguistic computing* **8** (4), 243–257, 1993.

[15] Hunston, S. *Corpora in applied linguistics*, Ernst Klett Sprachen, 2002.

[16] McFadden, T. *The position of morphological case in the derivation : A study on the syntax-morphology interface.* Ph.D. thesis, University of Pennsylvania, 2004.

[17] Müller, G. Free word order, morphological case, and sympathy theory. *Resolving Conflicts in Grammars: Optimality Theory in Syntax, Morphology, and Phonology* **11**, 1–9, 2002.

[18] Porter, M. F. An algorithm for suffix stripping. *Program: Electronic Library and Information Systems* **14** (3), 130–137, 1980.

[19] Sarkar, S. & Bandyopadhyay, S. Design of a rule-based stemmer for natural language text in Bengali. In *Proceedings of the IJCNLP Workshop on NLP for Less Privileged Languages.* 65–72, Hyderabad, India, 2008.

[20] Yarowsky, D. & Wicentowski, R. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics.* 207–216, Hong Kong, 2000.

[21] Wicentowski, R. Multilingual noise-robust supervised morphological analysis using the wordframe model. In *Proceedings of the 7th Meeting of the ACL Special Interest Group in Computational Phonology: Current Themes in Computational Phonology and Morphology.* 70–77, Barcelona, Spain, 2004.

[22] Majumder, P. *et al.* YASS: Yet another suffix stripper. *ACM Transactions on Information Systems* **25** (4), 2007.

[23] Paik, J. H. & Parui, S. K. A fast corpus-based stemmer. *ACM Transactions on Asian Language Information Processing* **10** (2), 1–16, 2011.

[24] Porter, M. F. Stemming algorithms for various European languages. http://snowball.tartarus.org/texts/stemmersoverview.html, 2012.

[25] Lovins, J. B. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics* **11** (1), 22–31, 1968.

[26] Lennon, M. *et al.* An evaluation of some conflation algorithms for information retrieval. *Journal of Information Science* **3** (4), 177–183, 1981.

[27] Harman, D. How effective is suffixing? *Journal of the American Society for Information Science* **42** (1), 7–15, 1991.

[28] Hull, D. A. Stemming algorithms: A case study for detailed evaluation. *Journal of the American Society for Information Science* **47** (1), 70–84, 1996.

[29] Oard, D. W. *et al.* CLEF '00 experiments at the University of Maryland: Statistical stemming and backoff translation strategies. In Peters, C. (ed.) *Cross-Language Information Retrieval and Evaluation.* 176–187, Springer, 2001.

[30] Dinçer, B. T. & Karaoglan, B. Stemming in agglutinative languages: A probabilistic stemmer for Turkish. In Yazici, A. & Sener, C. (eds.) *Computer and Information Sciences.* 244–251, Springer, 2003.

[31] Šnajder, J. & Bašic, B. String distance-based stemming of the highly inflected Croatian language. In *Proceedings of the International Conference on Recent Advances on Natural Language Processing.* 411–415, Borovets, Bulgaria, 2009.

[32] McNamee, P. *et al.* A language-independent approach to European text retrieval. In *Revised Papers from the Workshop of Cross-Language Evaluation Forum on Cross-Language Information Retrieval and Evaluation.* 129–139, London, UK, 2001.

[33] Kraaij, W. & Pohlmann, R. Viewing stemming as recall enhancement. In *Proceedings of the* 19^{th} *Annual International ACM SIGIR Conference on Research & Development in Information Retrieval.* 40–48, Zurich, Switzerland, 1996.

[34] Savoy, J. A stemming procedure and stopword list for general French corpora. *Journal of the American Society for Information Science* **50** (10), 944–952, 1999.

[35] Korenius, T. *et al.* Stemming and lemmatization in the clustering of Finnish text documents. In *Proceedings of the thirteenth ACM International Conference on Information and Knowledge Management.* 625–633, Washington, USA, 2004.

[36] Dolamic, L. & Savoy, J. Indexing and stemming approaches for the Czech language. *Information Processing & Management* **45** (6), 714–720, 2009.

[37] Larkey, L. S. *et al.* Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research & Development in Information Retrieval.* 275–282, Tampere, Finland, 2002.

[38] Rogati, M. *et al.* Unsupervised learning of Arabic stemming using a parallel corpus. In *Proceedings of the 41st Annual Meeting on ACL.* 391–398, Sapporo, Japan, 2003.

[39] Taghva, K. *et al.* Arabic stemming without a root dictionary. In *Proceedings of International Conference on Information Technology: Coding and Computing.* 152–157, Las Vegas, USA, 2005.

[40] Al-Shammari, E. T. & Lin, J. Towards an error-free Arabic stemming. In *Proceedings of the 2nd ACM Workshop on Improving non-English Web Searching.* 9–16, California, USA, 2008.

[41] Kudo, T. *et al.* Applying conditional random fields to Japanese morphological analysis. In *Proceedings of Conference on Empirical Methods on Natural Language Processing.* 230–237, Barcelona, Spain, 2004.

[42] Uchimoto, K. *et al.* The unknown word problem: A morphological analysis of Japanese using maximum entropy aided by a dictionary. In *Proceedings of Conference on Empirical Methods on Natural Language Processing.* 91–99, Cornell University, USA, 2001.

[43] Braschler, M. & Ripplinger, B. Stemming and decompounding for German text retrieval. In Sebastiani, F. (ed.) *Advances in Information Retrieval : 25th European Conference on IR Research*, 177–192, Springer, Pisa, Italy, 2003.

[44] Krovetz, R. Viewing morphology as an inference process. *Artificial Intelligence* **118** (1), 277–294, 2000.

[45] Ramanathan, A. & Rao, D. A lightweight stemmer for Hindi. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational*

Linguistics, on Computational Linguistics for South Asian Languages. 43–48, Budapest, Hungary, 2003.

[46] Pandey, A. & Siddiqui, T. J. An unsupervised Hindi stemmer with heuristic improvements. In *Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data.* 99–105, Singapore, 2008.

[47] Aswani, N. & Gaizauskas, R. Developing morphological analysers for South Asian Languages: Experimenting with the Hindi and Gujarati languages. In *Proceedings of the Seventh conference on International Language Resources and Evaluation.* 811–815, Malta, 2010.

[48] Paik, J. H. *et al.* GRAS: An effective and efficient stemming algorithm for information retrieval. *ACM Transactions on Information Systems* **29** (4), 1–24, 2011.

[49] Kumar, D. & Rana, P. Design and development of a stemmer for Punjabi. *International Journal of Computer Applications* **11** (12), 18–23, 2010.

[50] Majgaonker, M. M. & Siddiqui, T. J. Discovering suffixes: A case study for Marathi language. *International Journal on Computer Science and Engineering* **04**, 2716–2720, 2010.

[51] Ram, V. S. & Devi, S. L. Malayalam stemmer. In Parakh, M. (ed.) *Morphological Analysers and Generators.* 105-113, Mysore, India, 2010.

[52] Bora, L. S. *Asamiya Bhasar Ruptattva*, M/s Banalata, Guwahati, Assam, India, 2006.

[53] Goswami, U. *Asamiya Bhashar Vyakaran*, Mani Manik Prakash, Guwahati, Assam, India, 2001.

[54] Abbi. A. Reduplicative structures· A phenomenon of the south asian linguistic area. In *Oceanic Linguistics Special Publications*, 159 – 171, University of Hawaii Press, 1985

[55] Melucci, M. & Orio, N. A novel method for stemmer generation based on Hidden Markov Models. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management.* 131–138, New Orleans, USA, 2003.

[56] Saharia, N. *et al.* An improved stemming approach using hmm for a highly inflectional language. In *Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics.* 164–173, Samos, Greece, 2013.

[57] Viterbi, A. J. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transaction on Information Theory* **61** (3), 268–278, 1967.

[58] Creutz, M. & Lagus, K. Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0. Tech. Rep., Helsinki University of Technology, 2005.

[59] Creutz, M. & Lagus, K. Unsupervised discovery of morphemes. In *Morphological and Phonological Learning: Proceedings of the 6th Workshop of the ACL Special Interest Group in Computational Phonology.* 21–30, Philadelphia, USA, 2002.

[60] Sinha, K. P. *The Bishnupriya Manipuri Language*, Firma KLM Private Limited, Calcutta, India, 1982, first edn.

[61] Dasgupta, S. & Ng, V. Unsupervised word segmentation for Bangla. In *Proceedings of the 5th International Conference on Natural Language Processing.* 60–66, Hyderabad, India, 2007.

[62] Das, A. & Bandyopadhyay, S. Morphological stemming cluster identification for Bangla. In *Knowledge Sharing Event-1, Task-3: Morphological Analysers and Generators*, vol. 3, Mysore, India, 2010.

[63] Ekbal, A. & Bandyopadhyay, S. Bengali named entity recognition using support vector machine. In *Proceedings of Workshop on NER for South and South East Asian Languages in Collaboration with* 3^{rd} *International Joint Conference on Natural Language Processing.* 51–58, Hyderabad, India, 2008.

[64] Sharma, P. *et al.* Suffix stripping based NER in Assamese for location names. In *Proceedings of* 2^{nd} *National Conference on Computational Intelligence and Signal Processing.* 91–94, Guwahati, India, 2012.

[65] Prince, A. & Smolensky, P. Optimality theory: Constraint interaction in generative grammar. Tech. Rep., Rutgers University, NJ, USA, 1993.

[66] Ratnaparkhi, A. A simple introduction to maximum entropy models for natural language processing. Tech. Rep., University of Pennsylvania, Philadelphia, USA, 1997.

[67] Lafferty, J. *et al.* Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of International Conference on Machine Learning.* 282–289, Williamstown, USA, 2001.

[68] Brill, E. Transformation based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics* **21** (4), 543–565, 1995.

[69] Brants, T. TnT - a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing.* 224–231, Washington, USA, 2000

[70] Samuelsson, C. & Voutilainen, A. Comparing a linguistic and a stochastic tagger. In *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguisiics* 246–253, 1997.

[71] DINCER, T. *et al.* A suffix based part-of-speech for Turkish. In *Proceedings of 5th International conference on Information Technology: New Generations*, 2008.

[72] Ray, P. R. *et al.* Part of speech tagging and local word grouping techniques for natural language parsing in Hindi. In *Proceedings of the 1st International Conference on Natural Language Processing*, Mysore, India, 2003.

[73] Leech, G. *et al.* Claws 4: the tagging of the British National Corpus. In *Proceedings of the 15th conference on Computational linguistics.* 622–628, Association for Computational Linguistics, 1994.

[74] Ratnaparkhi, A. A Maximum Entropy model for part of speech tagging. In *Proceedings of the conference on Empirical methods in Natural Language Processing.* 133–142, Sydney, Australia, 1996.

[75] Nakagawa, T. *et al.* Unknown word guessing and part-of-speech tagging using support vector machines. In *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium.* 325–331, Tokyo, Japan, 2001.

[76] Schmid, H. Part-of-speech tagging with neural networks. In *Proceedings of the 15th conference on Computational linguistics.* 172–176, Association for Computational Linguistics, 1994.

[77] Màrquez, L. *et al.* Automatically acquiring a language model for pos tagging using decision trees , 1997.

[78] Schmid, H. Probabilistic part-of speech tagging using decision trees. In *Proceedings of international conference on new methods in language processing.* 44–49, Manchester, UK, 1994.

[79] Goldwater, S. & Griffiths, T. L. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics.* 744–751, Prague, Czech Republic, 2007.

[80] Yarowsky, D. & Ngai, G. Inducing multilingual PoS taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies.* 1–8, Pittsburgh, USA, 2001.

[81] Das, D. & Petrov, S. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.* 600–609, Portland, USA, 2011.

[82] Biemann, C. Unsupervised part-of-speech tagging employing efficient graph clustering. In *Proceedings of COLING/ACL 2006, Student research workshop.* 7–12, Sydney, Australia, 2006.

[83] Hajič, J. Morphological tagging: Data vs. dictionaries. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference,* 2000.

[84] Oflazer, K. & Kuruöz, I. Tagging and morphological disambiguation of Turkish text. In *Proceedings of 4th Conference on ANLP,* 1994.

[85] Shamsfard, M. & Fadaee, H. A hybrid morphology-based POS tagger for Persian. In *Proceedings of the the International Conference on Language Resources and Evaluation,* 2008.

[86] Ravi, S. & Knight, K. Minimized models for unsupervised part-of-speech tagging. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP.* 504–512, Singapore, 2009.

[87] Umansky-Pesin, S. *et al.* A multi-domain web-based algorithm for PoS tagging of unknown words. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters.* 1274–1282, Beijing, China, 2010.

[88] Georgiev, G. *et al.* Feature-rich part-of-speech tagging for morphologically complex languages: Application to Bulgarian. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics.* 492–502, Avignon, France, 2012.

[89] Shen, L. *et al.* Guided learning for bidirectional sequence classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics.* 760–767, Prague, Czech Republic, 2007.

[90] Habash, N. & Rambow, O. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics.* 573–580, Ann Arbor, USA, 2005.

[91] Dandapat, S. *et al.* A hybrid model for part-of-speech tagging and its application to Bengali. In *International Conference on Computational Intelligence.* 169–172, Istanbul, Turkey, 2004.

[92] Singh, S. *et al.* Morphological richness offsets resource demand-experiences in constructing a POS tagger for Hindi. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics : Poster*, 2006.

[93] Dalal. A. *et al.* Hindi part-of-speech tagging and chunking: A maximum entropy approach. *Proceeding of the NLPAI Machine Learning Competition* , 2006.

[94] Hellwig, O. SanskritTagger, a stochastic lexical and POS tagger for Sanskrit. In *Proceedings of First International Symposium on Sanskrit Computational Linguistics.* 37–46, Rocquencourt, France, 2007.

[95] Sastry, G. M. R. *et al.* A HMM based part-of-speech and statistical chunker for 3 Indian languages. In *Proceedings of IJCAI workshop on Shallow Parsing for South Asian Languages*, Hyderabad, India, 2007.

[96] PVS, A. & G, K. Part of speech tagging and chunking using conditional random fields and transformation based learning. In *Proceedings of the IJCAI workshop on Shallow Parsing for South Asian Languages.* 21–24, Hyderabad, India, 2007.

[97] Singh, T. D. & Bandyopadhyay, S. Morphology driven Manipuri PoS tagger. In *Proceedings of the IJCNLP Workshop on NLP for Less Privileged Languages.* 91–98, Hyderabad, India, 2008.

[98] Shrivastava, M. & Bhattacharyya, P. Hindi PoS tagger using naive stemming: Harnessing morphological information without extensive linguistic knowledge. In *Proceedings of the 5th International Conference on Natural Language Processing*, Pune, India, 2008.

[99] Manju, K. *et al.* Development of a PoS tagger for Malayalam - an experience. In *Proceedings of the International Conference on Advances in Recent Technologies in Communication and Computing.* 709–713, Kottayam, India, 2009.

[100] Sharma, S. & Lehal, G. Using Hidden Markov Model to improve the accuracy of Punjabi PoS tagger. In *Proceedings of the IEEE International Conference on Computer Science and Automation Engineering.* 697–701, Shanghai, China, 2011.

[101] Gupta, J. *et al.* A TENGRAM method based part-of-speech tagging of multi-category words in Hindi language. *Expert Systems with Applications* **38** (12), 15084–15093, 2011.

[102] Reddy, S. & Sharoff, S. Cross language PoS taggers (and other tools) for Indian languages: An experiment with Kannada using Telugu resources. In *Proceedings of the 5th Workshop on Cross Lingual Information Access.* 11–19, Chiang Mai, Thailand, 2011.

[103] Ekbal, A. & Saha, S. Simulated annealing based classifier ensemble techniques: Application to part of speech tagging. *Information Fusion* **14** (3), 288 – 300, 2012.

[104] Dandapat, S. Part-of-speech tagging and chunking with maximum entropy model. In *Proceedings of Workshop on Shallow Parsing for South Asian Languages*, Hyderabad, India, 2007.

[105] Saha, G. K. *et al.* Computer assisted Bangla words POS tagging. In *Proc. International Symposium on Machine Translation NLP and TSS*, 2004.

[106] Pammi, S. C. & Prahallad, K. POS tagging and chunking using Decision Forests. In *Proceedings of Workshop on Shallow Parsing for South Asian Languages*, Hyderabad, India, 2007.

[107] Goswami, G. C. *Asamiya Vyakaran Pravesh*, Bina Library, Guwahati, Assam, India, 2000.

[108] Bharati, A. *et al. Natural Language Processing: A Paninian Perspective*, Prentice-Hall, India, 1993.

[109] Dermatas, E. & Kokkinakis, G. Automatic stochastic tagging of natural language text. *Computational Linguistics* **21** (2), 137–163, 1995.

[110] Toutanova, K. *et al.* Feature-Rich part-of-speech tagging with a Cyclic Dependency Network. In *Proceedings of HLT-NAACL.* 173–180, 2003.

[111] Banko, M. & Moore, R. Part of speech tagging in context. In *proceedings of 20th international conference on Computational Linguistics*, 2004.

[112] Dandapat, S. & Sarkar, S. Part-of-speech tagging for Bengali with Hidden Markov Model. In *Proceedings of NLPAI ML workshop on Part of speech tagging and Chunking for Indian language*, 2006.

[113] T., P. R. *et al.* A text chunker and hybrid pos tagger for Indian languages. In *Proceedings of IJCAI workshop on Shallow Parsing for South Asian Languages*, Hyderabad, India, 2007.

[114] Rao, D. & Yarowsky, D. Part of speech tagging and shallow parsing of Indian languages. In *Proceedings of IJCAI workshop on Shallow Parsing for South Asian Languages*, Hyderabad, India, 2007.

[115] Ekbal, A. *et al.* Pos tagging using HMM and rule based chunking. In *Proceedings of Workshop on Shallow Parsing for South Asian Languages*, Hyderabad, India, 2007.

[116] Sag, I. A. *et al.* Multiword Expressions: A pain in the neck for NLP. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing.* 1–15, Mexico, 2002.

[117] Cortes, C. & Vapnik, V. Support vector machine. *Machine learning* **20** (3), 273–297, 1995.

[118] Lapata, M. The disambiguation of nominalizations. *Computational Linguistics* **28** (3), 357–388, 2002.

[119] Baldwin, T. Deep lexical acquisition of verb-particle constructions. *Computer Speech & Language* **19** (4), 398–414, 2005.

[120] Villavicencio, A. *et al.* Editorial: Introduction to the special issue on multiword expressions: Having a crack at a hard nut. *Computer Speech and Language* **19** (4), 365–377, 2005.

[121] Ramisch, C. *et al.* An evaluation of methods for the extraction of multiword expressions. In *Proceedings of the LREC Workshop-Towards a Shared Task for Multiword Expressions.* 50–53, 2008.

[122] Pearce, D. Synonymy in collocation extraction. In *Proceedings of the Workshop on WordNet and Other Lexical Resources.* 41–46, Pittsburg, USA, 2001.

[123] Kim, S. N. & Baldwin, T. Automatic identification of English verb particle constructions using linguistic features. In *Proceedings of the Third ACL-SIGSEM Workshop on Prepositions*. 65–72, 2006.

[124] Piao, S. S. *et al.* Measuring MWE compositionality using semantic annotation. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*. 2–11, Sydney, Australia, 2006.

[125] Oflazer, K. *et al.* Integrating morphology with multi-word expression processing in Turkish. In *Proceedings of the Workshop on Multiword Expressions: Integrating Processing*. 64–71, Barcelona, Spain, 2004.

[126] Grégoire, N. Design and implementation of a lexicon of Dutch multiword expressions. In *Proceedings of the Workshop on A Broader Perspective on Multiword Expressions*. 17–24, Prague, Czech Republic, 2007.

[127] Baldwin, T. *et al.* Multiword expressions: Some problems for Japanese NLP. In *Eighth Annual Meeting of the Association of Natural Language Processing*. 379–382, Keihanna, Japan, 2002.

[128] Zhang, Y. *et al.* Automated multiword expression prediction for grammar engineering. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*. 36–44, Sydney, Australia, 2006.

[129] Villavicencio, A. *et al.* Validation and evaluation of automatically acquired multiword expressions for grammar engineering. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 1034–1043, Prague, Czech Republic, 2007.

[130] Ramisch, C. *et al.* Multiword expressions in the wild?: the MWEToolkit comes in handy. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*. 57–60, Beijing, China, 2010.

[131] Attia, M. *et al.* Automatic extraction of Arabic multiword expressions. In *Proceedings of the Workshop on Multiword Expressions: from Theory to Applications*. 18–26, Beijing, China, 2010.

[132] Van de Cruys, T. & Moirón, B. Semantics-based multiword expression extraction. In *Proceedings of the Workshop on a Broader Perspective on Multiword Expressions*. 25–32, 2007.

[133] Vintar, Š. & Fišer, D. Harvesting multiword expressions from parallel corpora. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*. 1091–1096, Marrakech, Morocco, 2008.

[134] Duan, J. *et al.* A hybrid approach to improve bilingual multiword expression extraction. *Advances in Knowledge Discovery and Data Mining* 541–547, 2009.

[135] Moirón, B. & Tiedemann, J. Identifying idiomatic expressions using automatic word-alignment. In *Proceedings of the EACL Workshop on Multi-word expressions in a multilingual context*. 33–40, 2006.

[136] Okita, T. *et al.* Multi-word expression-sensitive word alignment. In *Proceedings of the Fourth International Workshop on Cross Lingual Information Access*, Beijing, China, 2010.

[137] de Caseli, H. M. *et al.* Alignment-based extraction of multiword expressions. *Language Resources and Evaluation* **44** (1), 59–77, 2010.

[138] Agarwal, A. *et al.* Automatic extraction of multiword expressions in Bengali: An approach for miserly resource scenario. In *Proceedings of International Conference on Natural Language Processing*. 165–174, Hyderabad, India, 2004.

[139] Dandapat, S. *et al.* Statistical investigation of Bengali noun-verb (NV) collocations as multi-word-expressions. 230–233. Mumbai, India, 2006.

[140] Kunchukuttan, A. & Damani, O. P. A system for compound noun multiword expression extraction for Hindi. In *Proceedings of the 6th International Conference on Natural Language Processing*. 20–29, Pune, India, 2008.

[141] Venkatapathy, S. & Joshi, A. Relative compositionality of multi-word expressions: a study of verb-noun (VN) collocations. In *Proceedings of the International Joint Conference on Natural Language Processing*. 553–564, Jeju Island, Korea, 2005.

[142] Mukerjee, A. *et al.* Detecting complex predicates in Hindi using PoS projection across parallel corpora. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*. 28–35, Sydney, Australia, 2006.

[143] Chakrabarti, D. *et al.* Hindi compound verbs and their automatic extraction. In *Proceedings of the International Conference on Computational Linguistics*. 27–30, Manchester, UK, 2008

[144] Sinha, R. M. K. Stepwise mining of multi-word expressions in Hindi. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World.* 110–115, Portland, USA, 2011.

[145] Chakraborty, T. *et al.* Semantic clustering: an attempt to identify multiword expressions in Bengali. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World.* 8–13, Portland, USA, 2011.

[146] Baldwin, T. *et al.* Road-testing the English resource grammar over the British National Corpus. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation.* 2047–2050, Lisbon, Portugal, 2004.

[147] Finin, T. W. The semantic interpretation of nominal compounds. In *Proceedings of the First Annual National Conference on Artificial Intelligence.* 3–10, 1980.

[148] Buckeridge, A. M. & Sutcliffe, R. F. Disambiguating noun compounds with latent semantic indexing. In *Proceedings of Second International Workshop on Computational Terminology.* 1–7, Taipei, Taiwan, 2002.

[149] Hook, P. E. *The compound verb in Hindi.* Ph.D. thesis, University of Pennsylvania, 1973.

[150] Sinha, R. M. K. Mining complex predicates in Hindi using a parallel Hindi-English corpus. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications.* 40–46, Singapore, 2009.

[151] Begum, R. *et al.* Identification of conjunct verbs in Hindi and its effect on parsing accuracy. *Computational Linguistics and Intelligent Text Processing* 29–40, 2011.

[152] Ramshaw, L. A. & Marcus, M. P. Text chunking using transformation-based learning. In *Proceedings of the Third Association for Computational Linguistics Workshop on Very Large Corpora.* 82–94, Cambridge, USA, 1995.

[153] Steedman, M. *The Syntactic Process,*, MIT Press, 2000.

[154] Tesniére, L. *Éléments de syntaxe structurelle*, Paris, Klincksieck, 1959.

[155] Bresnan, J. & Kaplan, R. Lexical-functional grammar: A formal system for grammatical representation. In Bresnan, J. (ed.) *The Mental Representation of Grammatical Relations.* 173–281, Cambridge, Massachusetts, 1982.

[156] Joshi, A. K. *An introduction to tree adjoining grammars*, vol. 1, John Benjamins Publishing Company, 1987.

[157] Pollard, C. J. & Sag, I. A. *Head-driven Phrase Structure Grammar*, University of Chicago Press, 1994.

[158] Ratnaparkhi, A. *et al.* A maximum entropy model for parsing. In *Proceedings of the International Conference on Spoken Language Processing.* 803–806, 1994.

[159] Bharati, A. *et al.* A two-stage constraint based dependency parser for free word order languages. In *Proceedings of the International Conference on Asian Language Processing*, Chiang Mai, Thailand, 2008.

[160] Koo, T. *et al.* Simple semi-supervised dependency parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics : Human Language Technology Conference*, Ohio, USA, 2008.

[161] Nivre, J. *et al.* The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the Shared Task Session of EMNLP-CoNLL.* 915–932, Prague, 2007.

[162] Goldberg, Y. & Elhadad, M. An efficient algorithm for easy-first non-directional dependency parsing. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics.* 742–750, California, 2010.

[163] Nivre, J. *et al.* Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering* **13** (2), 95–135, 2007.

[164] Buchholz, S. & Marsi, E. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the 10th Conference on Computational Natural Language Learning.* 149–164, New York City, USA, 2006.

[165] Sagae, K. & Lavie, A. A classifier-based parser with linear run-time complexity. In *Proceedings of the Ninth International Workshop on Parsing Technologies (IWPT),*. 125–132, Vancouver, Canada, 2005

[166] Nivre, J. & Scholz, M. Deterministic dependency parsing of English text. In *Proceedings of COLING 2004.* 64–70, Geneva, Switzerland, 2004.

[167] Yamada, H. & Matsumoto, Y Statistical dependency analysis with support vector machines. In *Proceedings of the Eight International Workshop on Parsing Technology (IWPT)*, 2003.

[168] Bharati, A. *et al.* Paninian grammar framework applied to English. Tech. Rep., IIT Kanpur, Department of CSE, 1996.

[169] Bharati, A. & Sangal, R. Parsing free word order languages in the Paninian framework. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*. 105–111, Columbus, USA, 1993.

[170] Husain, S. Dependency parsers for Indian languages. In *Proceedings of ICON NLP Tools Contest: Indian Language Dependency Parsing*, Hyderabad, India, 2009.

[171] Collins, M. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania, 1999.

[172] Collins, M. *Computational Linguistics* **29** (4), 589–637, 2003.

[173] Nivre, J. & Nilsson, J. Pseudo-projective dependency parsing. In *Proceedings of the 43rd Annual Meeting of the ACL*. 99–106, Ann Arbor, 2005.

[174] Nivre, J. *et al.* Memory-based dependency parsing. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL)*. 49–56, Boston, USA, 2004.

[175] Cheng, Y. *et al.* Machine learning-based dependency analyzer for Chinese. *Journal of Chinese Language and Computing* **15** (1), 13–24, 2005.

[176] Marinov, S. & Nivre, J. A data-driven dependency parser for Bulgarian. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories*. 89–100, Barcelona, Spain, 2005.

[177] Eryiğit, G. & Oflazer, K. Statistical dependency parsing of Turkish. In *Proceedings of the 11th Conference of the European Chapter of the ACL*. 89–96, Trento, Italy, 2006.

[178] Collins, M. *et al.* A statical parser for Czech. In *Proceedings of the 37th Annual Meeting - Association for Computational Linguistics*. 505–512, Maryland, USA, 1999.

[179] Cowan, B. & Collins, M. Morphology and reranking for the statistical parsing of Spanish. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. 795–802, Morristown, USA, 2005.

[180] Bikel, D. M. Design of a multi-lingual, parallel-processing statistical parsing engine. In *Proceedings of the second international conference on Human Language Technology Research*. 178 –182, San Diego, USA, 2002.

[181] Dubey, A. & Keller, F. Probabilistic parsing for german using sister-head dependencies. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics.* 96–103, Sapporo, Japan, 2003.

[182] McDonald, R. & Pereira, F. Online learning of approximate dependency parsing algorithms. In *Proceedings of the 11th Conference of the European Chapter of the ACL.* 81–88, Trento, Italy, 2006

[183] Eisner, J. M. Three new probabilistic model for dependency parsing: An exploration. In *Proceedings of 16th International Conference on Computational Linguistics (COLING).* 340–345, Copenhagen, Denmark, 1996.

[184] Bouma, G. & van Noord, G Constraint-based categorial grammar. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics.* 147–154, Las Cruces, USA, 1994.

[185] Hoffman, B A CCG approach to free word order language. In *Proceedings of the 30th annual meeting on Association for Computational Linguistics.* 300–302, Deleware, USA, 1992.

[186] Vempaty, C. *et al.* Issues in analyzing Telugu sentences towards building a Telugu treebank. In Gelbukh, A. (ed.) *Computational Linguistics and Intelligent Text Processing*, LNCS, 50–59, 2010

[187] Eades, D. *et al.* A Paninian approach to parsing relative clauses in Hindi and Arabic. In *Proceedings of the International Conference on Natural Language Processing.* 33–42, New Delhi, India, 2004.

[188] Attardi, G. *et al.* Bengali parsing system at ICON NLP tool contest 2010. In *Proceedings of ICON NLP Tools Contest: Indian Language Dependency Parsing.* 15–19, Kharagpur, India, 2010.

[189] Ghosh, A. *et al.* Dependency parsing of Indian languages with DeSR. In *Proceedings of ICON NLP Tools Contest: Indian Language Dependency Parsing.* 20–24, Kharagpur, India, 2010.

[190] Kolachina, P. *et al.* Grammar extraction from treebanks for Hindi and Telugu. In *Proceedings of The 7th International Conference on Language Resources and Evaluation.* 3803–3810, 2010.

[191] Kesidi, S. R. *et al.* A two stage constraint based hybrid dependency parser for Telugu. In *Proceedings of ICON NLP Tools Contest: Indian Language Dependency Parsing.* 25-31, Kharagpur, India, 2010.

[192] Dryer, M. S. *Order of Subject, Object and Verb*, Max Planck Institute for Evolutionary Anthropology, Leipzig, 2013. URL `http://wals.info/chapter/81`.

[193] Rahman, M. *et al.* Parsing of part-of-speech tagged Assamese texts. *International Journal of Computer Science Issues* **6** (1), 2009.

[194] Sleator, D. & Temperley, D. Parsing English with a link grammar. In *Proceedings of the Third International Workshop on Parsing Technologies*, Tokyo, Japan, 1993.

[195] Nivre, J. Dependency grammar and dependency parsing. Tech. Rep., Växjö University, 2005.

[196] Chang, C.-C. & Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* **2** (3), 27:1–27:27, 2011.

[197] Fan, R.-E. *et al.* Liblinear: A library for large linear classification. *The Journal of Machine Learning Research* **9**, 1871–1874, 2008.

[198] McDonald, R. *et al.* Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*. 216–220, New York, USA, 2006.

[199] Rambow, O. *et al.* A dependency treebank for English. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, Canary Islands, Spain, 2002.

[200] Bosco, C. *et al.* Building a treebank for Italian: A data-driven annotation schema. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, Athens, Greece, 2000.

[201] Bosco, C. Multiple-step treebank conversion: from dependency to Penn format. In *Proceedings of the Linguistic Annotation Workshop*. 164–167, Prague, Czech, 2007.

[202] Bos, J. *et al.* Converting a dependency treebank to a categorial grammar treebank for Italian. In *Proceedings of the Eighth International Workshop on Treebanks and Linguistic Theories*. 27–38, Milan, Italy, 2009.

[203] Böhmová, A. *et al.* The Prague dependency treebank. In *Treebanks*, 103–127, Springer, 2003.

[204] Hajic, J. *et al.* Prague Arabic dependency treebank: Development in data and tools. In *Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools*. 110–117, Cairo, Egypt, 2004.

[205] Mikulová, M. & Štepánek, J. Annotation procedure in building the Prague Czech-English dependency treebank. In *Proceedings of Fifth International Conference on NLP, Corpus Linguistics Corpus Based Grammar Research.* 25–27, Smolenice/Bratislava, Slovakia, 2009.

[206] Ide, N. *et al.* An XML-based encoding standard for linguistic corpora. In *Proceedings of the Second International Conference on Language Resources and Evaluation.* 825–830, Athens, Greece, 2000

[207] Buch-Kromann, M. & Korzen, I The unified annotation of syntax and discourse in the Copenhagen dependency treebanks. In *Proceedings of the Fourth Linguistic Annotation Workshop.* 127–131, Uppsala, Sweden, 2010.

[208] Liu, H. & Huang, W. A Chinese dependency syntax for treebanking. In *Proceedings of the 20th Pacific Asia Conference on Language, Information, Computation.* 126–133, Wuhan, China, 2006.

[209] Van der Beek, L. *et al.* The Alpino dependency treebank. *Language and Computers* **45** (1), 8–22, 2002.

[210] Haverinen, K. *et al.* Building the essential resources for Finnish: the Turku dependency treebank. *Language Resources and Evaluation* 1–39, 2013.

[211] Brants, S. *et al.* The TIGER treebank. In *Proceedings of the workshop on Treebanks and linguistic theories.* 24–41, Sozopol, Bulgaria, 2002.

[212] Bharati, A. *et al.* SSF: Shakti Standard Format guide. Tech. Rep. Report No: IIIT/TR/2009/85, International Institute of Information Technology, Hyderabad, India, 2009.

[213] Begum, R. *et al.* Dependency annotation scheme for Indian languages. In *Proceedings of The Third International Joint Conference on Natural Language Processing,* Hyderabad, India, 2008.