Unsupervised Learning of Morphology of a Highly Inflectional Language

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

Utpal Sharma
Registration No. 014 of 2006

School of Engineering
Department of Computer Science and Engineering
Tezpur University
December 2006

Abstract

The primary information content of a natural language expression is partly determined by the meanings of the individual elements of the expression, and partly by the structure of the expression. Morphology is a major structural phenomenon in highly inflectional languages and plays a significant role in the formation of meaningful expressions. As a structural phenomenon the morphological behaviour of a language can be studied, and possibly acquired, by considering expressions in a sufficiently large text corpus of the language, in an unsupervised way. Use of effective methods for unsupervised learning of morphology for highly inflectional languages such as Assamese, which have large user bases, but do not receive much attention of computational linguistic research, can be quite beneficial.

Unsupervised approaches for acquisition of morphology of natural languages from raw text corpora have been proposed by different researchers. We observe that issues specific to particular languages, their scripts and the encoding schemes, must be addressed to make unsupervised acquisition effective. In this thesis, we focus on the Assamese language, which belongs to the Indic branch of the Indo-Aryan family of languages. Suffixation is the major structural phenomenon in Assamese. Using an unsupervised approach and several heuristics developed to address the issues not covered by general unsupervised acquisition models, the suffixation behaviour is learnt from a raw text corpus. Simultaneously, a morphological lexicon based on the input corpus is built. Then we have developed an approach to effectively use the acquired knowledge and the lexicon for morphological analysis of unseen texts. Impressive results have been observed in trials over texts from diverse domains. We further discuss the possibility of classifying words into morphological categories based on their observed suffixation behaviour. Starting from some simple ideas theoretically sound methods have been proposed and tested.

Keywords — morphology, Assamese, unsupervised acquisition, highly inflectional, Indic language, lexicon, word classification, morphological analysis

TEZPUR UNIVERSITY

Certificate

This is to certify that the thesis entitled "*Unsupervised learning of morphology of a highly inflectional language*" submitted to the Tezpur University in the Department of Computer Science and Engineering under the School of Engineering in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy in Computer Science is a record of research work carried out by Mr. Utpal Sharma under our personal supervision and guidance.

All help received by him from various sources has been duly acknowledged.

No part of this thesis has been reproduced elsewhere for award of any other degree.

Jugal Kumar Kalita

(Jugal K Kalita)
Associate Professor
University of Colorado
Colorado Springs
Colorado USA

Date: 11/22/2006
Place: Colorado Springs

Rajib Kr Das

(Rajib Kumar Das)
Reader
Department of Computer Science and Engineering
Tezpur University
Napaam, Assam, India

Date: 27/12/2006
Place: Tezpur

Declaration

I, Utpal Sharma, hereby declare that the thesis entitled "*Unsupervised learning of morphology of a highly 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.

Utpal Sharma

(Utpal Sharma)

Acknowledgements

The work described in this thesis has been possible because of the whole-hearted co-operation and support that I received from many persons and organisations. I hereby express my gratitude to all of them.

My supervisors Dr Rajib Das and Dr Jugal Kalita have been the driving force all along my work. Besides giving me technical guidance they impressed upon me the significance of the work that I had undertaken, and encouraged me throughout the exercise. Dr Jugal Kalita most generously provided me, from the USA, much literature that I needed for the work.

Prof Dilip Saikia, Prof Malayananda Dutta and Prof Dhruba Bhattacharyya of Tezpur University encouraged me and provided very important procedural guidance at different stages of my work.

Dr Shyamanta Hazarika, Dr Smriti Sinha and Nityananda Sarma of Tezpur University helped me in various ways in preparation of the thesis.

Prof Arun K Pujari of University of Hyderabad, and Mr Jatin Golani of Mumbai provided me some very useful literature.

My colleagues Sarat Saharia, S I Singh, Bhabesh Nath, Lipika Deka, Bulendra Gogoi, Kamakhya Gupta, Bhogeswar Bora, Rajib Goswami, Pranabjyoti Dutta, Debasish Das, Dhiraj Sharma, Ajay Sharma and Nitul Dutta helped me in various ways, and always encouraged me. Other members of the technical and administrative staff of the Department of Computer Science and Engineering and the Computer Centre too, helped create a wholesome environment to work in.

Without the care, support and patience of my parents I would not have been able to carry out this work.

Part of this work was carried out under a project funded by the All India Council of Technical Education, New Delhi, under their Research Promotion Scheme.

The Assamese texts in electronic form used in this my work were obtained from two sources – the Emille corpus from the website of the University of Lancaster, and the Internet edition of the Assamese daily Asomiya Pratidin.

The Assamese fonts used in this thesis have been produced using the LaTeX compatible *bwti* package available in the website of the Indian Institute of Science, Bangalore, after minor modifications.

Utpal Sharma

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Computers today possess enormous computing power at affordable costs. This power has been put to diverse information processing tasks. In these tasks the input, output or stored "information" is encoded in some format (scheme) suitable for the computer. These schemes of information encoding are some kind of "languages". Many information processing tasks of computers are over some restricted domain, and hence there are domain specific languages for representing information. Gradually, the need for computers to be able to process information in human languages has been felt. Human languages, also called natural languages, are highly versatile systems of encoding information. These can capture information of various domains. Such power of expression is achieved only with a fair amount of complexity. To enable a computer to process information in human languages, the language needs to be appropriately "described" to the computer, *i.e.*, the language needs to be "modelled".

Despite the complexities inherent in natural languages, human beings can learn such languages without any explicit instruction. Particularly, a child acquires his/her mother tongue primarily through exposure to "expressions" in that language. This means the human brain automatically acquires the necessary details of a language from evidence.

In this thesis we present an approach for acquisition of morphology of a highly inflectional language into a computational model, and our experiments for the Assamese language. It is basically an unsupervised learning approach,

particularly suitable for languages with a rich concatenative morphology. We have discussed the issues that arise in such an exercise. Broadly, our work covers three tasks - acquire the morphology of Assamese from a raw (unannotated) text corpus, use the morphological knowledge to analyse words in texts provided as input, and classify words according to their morphological behaviour. As part of the first task we also build a *morphological lexicon* (a list of root words with certain morphological attributes).

A prime motivation behind this work is to eventually develop a computational linguistic model of Assamese. Though Assamese is an important and a national language of India, little computational work has been done so far for this language. Ours is one of the first efforts in this regard and can be considered pioneering. There are many such languages for which it is very important to have a suitable but inexpensive computational acquisition process. These languages receive very little attention of computational linguistic research both in terms of availability of funds and number of researchers. We however do not claim that our approach is a solution for all such languages. Different languages have characteristics that require individual research attention.

1.1 Summary

The knowledge that needs to be modelled for an NLP system has been stratified roughly as- phonology/script, morphology, syntax, semantics, discourse, pragmatics and world knowledge. Till now a major demand for NLP is for processing written expressions, where phonology is not of concern. So to build a computational model one has to start from script and morphology. Script is primarily a system of visual symbols marked on some surface to denote some linguistic expression. In computational domain a system of numeric encoding of the visual script symbols is also important for internal representation of texts in a computer. The next level of knowledge in an NLP system is morphology. More precisely, this level of knowledge comprises vocabulary and morphology. Our efforts have been focused on acquisition of this knowledge for Assamese.

In a natural language vocabulary is best considered as an "open" set, whereas

morphology is a *finite* set of rules governing formation of words from more basic meaning-conveying elements. Vocabulary is an open set primarily because the *domain of information* keeps on changing– existing domains expand, and new domains arise. By domain of information we mean the entities and concepts that are covered in an expression. Apart from this, many languages have rich morphology, which makes many additional words possible from the basic forms of words. A human user of a language actually learns the morphology of the language "completely" and a "relevant" subset of the vocabulary. His vocabulary grows as and when needed. In our work we have tried to follow this strategy, that is, acquire the morphology of Assamese and build a lexicon of words encountered in the training stage. Further, the system should be able to recognise new words in test inputs.

Morphology is mainly manifested as affixation and compound formation. In Assamese the predominant morphological phenomenon is suffixation. Suffixation occurs both as inflection and derivation, and is usually simple concatenative. Though prefixes are also fairly frequent, the behaviour of the resultant words are often independent. Compounds are comparatively less frequent and can be treated as independent words too. In our work we have basically tried to acquire the suffixational morphology of Assamese.

We have implemented a simple algorithm that analyses a raw corpus of Assamese text and figures out the suffixation patterns that are in use. While this involves identification of frequently occurring trailing parts in words, it requires some carefully worked out filtering criteria to eliminate noise and retain only true suffix occurrences. The filtering criteria are general and have little to do with a particular language. Further, a morphologically rich language like Assamese allows multiple suffixes to occur consecutively after the root in a word. A good analysis must identify such cases. Not all arbitrary sequences of suffixes are valid. We have identified the particular suffix-sequences occurring in the input corpus. In particular, from a raw training corpus of about 300,000 words, we have identified 381 suffixes including 67 that are invalid. Of these 102 are actually words that attach with some base to form compounds, and 76 are composite suffixes, *i.e.*, they comprise multiple suffixes in sequences. In Assamese the distinction between of individual suffixes, suffix-sequences, and compound-parts

is often unclear. We have separately identified over 1700 suffix-sequences.

For effective morphological analysis it is very useful to be equipped with good morphological lexicon, *i.e.*, a list of words, preferably with some indication of their likely morphological behaviour. Hence, as we process the training corpus to learn the morphology, we also accumulate the base forms of words that we identify, along with their respective suffixation behaviour, in the form of a morphological lexicon. The lexicon that we have built from the raw corpus has over 15000 bases in about 26500 entries.

Since the vocabulary of a language is not finite, a lexicon cannot be exhaustive. Moreover, in our case the lexicon incorporates the vocabulary of only the training corpus, and the morphological information in it is not perfect since it is acquired by an unsupervised method. We have developed an approach that uses the acquired morphological knowledge and the lexicon to carry out morphological analysis of words in input texts. That is, the words in the input text are decomposed into the constituent root and suffix parts. We have achieved a precision of around 90% and recall of around 85% on average for input texts of diverse domains.

Words in a language are classified according to the roles they play in expressions. The commonly identified categories of words are nouns, pronouns, verbs, adjectives, *etc.* The morphological behaviour of words broadly depends on their categories, and it should be possible to guess the category of words by considering their morphological behaviour in a given input. In case of a an inflectional language where the predominant morphological phenomenon is suffixation, this means that it should be possible to guess the categories of words from the suffixation evidence. However, it is generally seen that there is lot of ambiguity in the evidence since within the same category the morphological behaviour words may vary, and words of different categories may display identical morphological behaviour to certain extent. In addition, the suffixation evidence that we extract using unsupervised methods is often sparse and contains noise. We have developed some methods based on set theory that can be used in these conditions, to classify words. The categories of words are not pre-defined; rather they are identified by considering the morphological evidence extracted from

the training corpus. These categories are more fine-grained than the commonly known word categories, and govern the morphological behaviour of words more closely.

1.2 Outline

In Chapter 2 we take a brief look at views on language acquisition by humans, and present our own insight of the problem. We consider the general nature of human languages and writing systems, with the objective of language acquisition, and state that morphology is a core component of a human language. As a structural phenomenon, it is possible to acquire morphology from written forms of linguistic expressions.

In Chapter 3 we discuss the salient features of the Assamese language. Though we stress that our language acquisition approach do not rely on the linguistic details, still it is accepted that certain linguistic features determine the effectiveness of a particular acquisition approach.

In Chapter 3 we also discuss the script encoding scheme that we have used. Unlike English, for which the Roman script has been so *naturalised* for use in a computer, for Assamese there is much standardisation to be done regarding the encoding of the script and the fonts to be used. We have had to work in an environment where different sources of corpus use different encoding schemes. So we have defined our own script encoding scheme, which has certain important advantages for our purpose. To keep compatibility with the different corpus sources, we have developed the necessary transliteration software.

In Chapter 4 we discuss our approach for identification of suffixes in a language from a raw text corpus of Assamese. It starts with a simple word stemming algorithm and goes on to describe certain important filtering criteria to reduce errors. We also discuss the creation of a morphological lexicon by accumulating word information from the input corpus. The set of suffixes, suffix-sequences and the morphological lexicon are the major deliverables of our work.

In Chapter 5 we discuss an exercise of morphological analysis of the words in an input Assamese text.

In Chapter 6 we discuss some possible methods for identifying classes of words based on suffix evidence. We start with simple intuitive approaches and move on to tackle issues that arise in the task. We finally advocate a set theoretic approach that fits the problem at hand. The set of classes identified can be considered as a morphological model of the language.

In Chapter 7 we conclude by summarising our major achievements and discussing some possible future work in this line.

In Appendix A we provide a brief overview of the Assamese alphabet and the transliteration scheme that we use, for the benefit of interested but unaccustomed readers.

In Appendix C we provide the implementation outlines of some of the procedures mentioned in the main chapters.

Chapter 2

The Problem of Language Acquisition

Natural language, or human language, is one of the most important tools that make humans more powerful than other living beings. It is unlike most other traits of humans in the sense that there is lot of variations in the languages that different groups of people use and these variations are social rather than strictly racial or locational. On one hand, each natural language contains lot of properties that apparently makes it a hard to acquire. On the other, children acquire their first language, whichever it is, so smoothly and rather fast. Early on in a person's life language becomes his strongest and most visible intellectual capability. From a formal standpoint, a natural language contains lot of ambiguities. But in practice this creates difficulty only rarely. Rather, often these ambiguities are utilized to achieve certain objectives. These factors have drawn attention of researchers to linguistics and language acquisition for a long time. After the emergence of the idea of use of natural languages in computers in the last century, linguistics and language acquisition has acquired a computational dimension.

According to the *Oxford Advanced Learner's Dictionary of Current English*, language is a "system of sounds, words, patterns, *etc.* used by humans to communicate thoughts and feelings" ([23]). It also describes language as a "manner of expressing oneself", "system of signs, symbols, gestures, *etc.* used for conveying information" and "system of coded instructions used in programming"

(in the domain of computing). We can arrive at a generalised definition to say that "language is a system for representing information, comprising a set of elementary parts each conveying some *meaning* and a set of rules for combining the elementary parts to convey some composite information". In the domain of computing, language is used to represent data and logic for computations. Humans use *natural language* for communication and recording information.

2.1 Views on language acquisition

As systems of representation of diverse kinds of information, human languages have been a subject of considerable interest. The easy and *natural* acquisition of such languages by children who do not have any other significant means of communication makes language acquisition too, interesting. One of the early schools of thought regarding language acquisition by children is that of *empirism*. According to this, linguistic knowledge is conceived as a product of living in an environment, a series of messages of "nurture" transmitted by other individuals and one's surrounding culture ([17, 35]). It is acquired through perceptions. Jean Piaget deemed this idea insufficient. He emphasized that there must also be *schemes of action*. Knowledge proceeds from action, and all action that is repeated or generalised through application to new objects engenders by this very fact a "scheme" that is, a kind of practical concept ([33]). Piaget proposed the concept of *genetic epistemology*, according to which knowledge can be constructed only through certain inborn modes of processing available to the young child and the actual characteristics of physical objects and events. Human linguistic capacities can be considered as a product of general "constructed" intellectual development. Piaget emphasizes on adaptation, assimilation, homeostasis, and autoregulation. Through these there is a *transfer of order* or *transfer of structure* from the environment to the organism ([34]).

On the other hand, Noam Chomsky proposed the concept of *innatism* of linguistic knowledge ([9, 35]). According to him, knowledge is largely inborn, part of the individual's birthright, a form of innate ideas existing in the realm of "nature". In other words, human linguistic capacities are a highly specialized

part of human genetic inheritance, largely separate from other human faculties and more plausibly viewed as a kind of innate knowledge that only has to unfold. More generally, the laws of order, including cognitive and linguistic, are inborn in an organism and are imposed by it upon the perceptible world. These laws are not derived from that world. They are species-specific and invariant over time, individuals and cultures. Linguistic studies attempt to characterize in depth and detail the internal structure in an organism through adequate abstractions from empirical observations ([34]). In course Chomsky proposed the idea of *Universal Grammar (UG)* ([10]): 'the system of principles, conditions, and rules that are elements or properties of all human languages... the essence of human language'. UG is the inborn linguistic knowledge that all humans possess irrespective of which language one speaks. UG was first described specifically as *Government/Binding (GB)* theory. Subsequently, the *principles and parameters theory* became more popular in describing UG. The central claim of this theory is that language knowledge consists of principles universal to all languages and parameters that vary from one language to another. Principles are such as, structure dependency, a head parameter in each phrase, and the projection principle. More recently, the *Minimalist Programme* has gained attention in this line ([10, 24]). The overall aim remains making statements about human languages that are as simple and general as possible. In particular, minimalism emphasizes *Full Interpretation and Economy of representation and derivation.* Full interpretation means that all elements in the structure of a sentence plays some *essential* role and must be interpreted in some way. The principle of Economy means that all representations and processes used to derive them should be as economical as possible.

For computational acquisition of language, both constructivism of Piaget, and innatism of Chomsky, requires us to tackle the issue of developing the fundamental structure of linguistic competence in the computer. If we adopt constructivism we shall have to suitably model and implement the "intelligence" that would assimilate the linguistic knowledge from the environment. If we adopt innatism we shall have to model and implement the innate linguistic competence that exist in humans at birth.

From the computational perspective another view that has developed is the

connectionist view. Connectionist NLP means using neural networks for NLP tasks.

2.2 Evolution of languages and linguistic capabilities

On our part, we shall look at the possible evolution of languages to develop an insight into its acquisition. However, it must be understood that in general NLP looks at the language acquisition process in human only as a possible model for adoption, not the sole one. In fact, our understanding of language acquisition in humans so far is at best incomplete, and cannot yet serve as a complete model for language acquisition for a computer. For parts of the overall NLP task, theories proposed for language acquisition in humans can be *realised fruitfully* in a computer, and hence may be used in NLP. For the remaining portion of NLP, purely computational ideas are used. In course of its evolution each natural language develops its own peculiarities, which make certain aspects of acquisition more difficult than others. The approach for acquisition of a particular language may be tailored specific to the peculiarities of that language.

Living organisms have a tendency to strive for their *well-being.* For plants this "well-being" basically implies growth and survival. This explains the growth of certain plants in such a way that would ensure more sunlight, better acquisition of nutrients, *etc.* For animate beings, well-being also means *security.* This explains their tendency of aggressiveness and self-defence in adverse environments. The quest for well-being of these organisms gives rise to their social behaviour. Social behaviour can be seen in the form of care of children by parents and attachment between members of a group. These traits enhance the survivality of these beings. Animate beings, particularly humans, have *memory* and *intelligence.* They are aware of the present environment as well as the past, and can figure out the future.

Depending on the quantity of memory and intelligence the idea of *well-being* attains different scales. Intelligence helps convert the memory contents into *knowledge.* Further, intelligent beings observe that sharing of *individual*

knowledge leads to enhancement of a pool of knowledge, and this in turn leads to more effective collective steps towards well-being of the group. Sharing of individual knowledge is achieved through some means of *communication*.

The means of communication between members of a society depends on their intelligence and their physiological capabilities. Communication essentially requires mutually comprehensible use of gestures and symbols by the communicating parties. Some communication in animals other than humans takes place through physical gestures such as licking, movement of body parts, *etc.*, and a few different kinds of sounds produced by the mouth, wings, *etc.* Rarely, if ever, multiple gestures or sounds are combined in any constructive way. The information that can be conveyed by these means is limited. Owing to low intelligence levels, and possibly, small memory, such animals generally have very limited domains of information to convey. Their communication needs arise from their instantaneous conditions such as hunger, fear, pleasure, pain, *etc.* Their familiarity with their languages is probably part of their inborn intelligence and physiological characteristics. With their level of intelligence they do not form ideas that are not obvious, and hence there is hardly any need for any other communication. Moreover, they lack the physiological capability to produce the variety of sounds (or any other signals) required for more versatile communication.

Humans have the rather unique *combination of high intelligence and the ability to produce (and recognize) a wide range of distinct sounds.* Due to a higher level of intelligence, the perception of the environment is deeper leading to a richer assortment of knowledge. Humans form non-trivial ideas and schemes for the future. Hence they need a means of communication more powerful than simple physical gestures or use of single words. Over time they have used their intelligence and their physiological capability to produce different types of vocal sounds to evolve a means of communication using different sounds, which we now call a *human language.* These languages have elaborate systems of rules to encode and decode information of arbitrary complexity and from different domains.

The *existence of the multiple human (natural) languages instead of a single universal language*, indicates that unlike other animate beings, the familiarity with a *particular* language is not part of their inborn intelligence. Till an infant

learns a natural language, his/her language comprises simple gestures such as crying and laughing. This language is probably universal across societies and cultures, and part of the inborn intelligence of humans. Gradually in the social environment that a child grows, he/she acquires one or more natural languages.

It is natural to suppose that neither the capability to produce the wide range of sounds nor any of the present human languages emerged in a single step of evolution. The evolution paths of different languages have not been very similar. The factors for evolution of languages include intellectual contributions and maturation, convenience of usage, the domains of usage, effect of other languages, and availability of more useful alternative languages. Convenience of usage is generally that of spoken form of the language. Individual sound elements undergo modifications when they occur with certain other sound elements. Similarly, words gradually assume forms, which are more convenient to utter.

2.3 Written form of a language

The spoken and written forms of a natural language are two different information *encoding schemes*. Speech is the primary form of a natural language. Written form is the secondary form. In most writing systems the latter actually encodes the former, *i.e.*, the written form simply denotes sounds that in turn convey the information. This encoding can cause some loss of information too. There are some writing systems where the written symbols directly represent entities and concepts. Such writing systems probably evolved from the use of pictures to express thoughts. Unlike the spoken form of a language, humans do not naturally acquire the written form, as part of their growing up.

The writing systems of languages generally use a set of symbols, *the alphabet*. Different writing systems use the symbols differently. In some, the symbols represent sounds, or phones, strictly, while in others the mapping between symbols and sounds is not very strict. In some, the symbols represent entities or concepts rather than sounds. In speech, elements like pause and tones are often used to achieve certain effects Generally, there are no exactly similar devices in writing systems. Punctuation symbols and sometimes, special fonts (especially

in printed text), *etc.* are employed for such purposes.

2.4 Language as a framework for representing information

Language is used for encoding information for the purpose of communication The information is not necessarily sequential. For example, consider a geometrical figure of a circle inside a square whose side is as long as the diameter of the circle. This figure is not sequential, nor the information contained in it. But it is possible to describe the diagram using a natural language expression, say, English sentences. These expressions are *sequential* in nature, *i.e.*, each expression is a sequence of words. The order in which the portions of the diagram are described may not be important provided the descriptions are correct and do not contradict any information in the diagram. Natural languages usually have mechanisms to express *relationships* between *objects* and *attributes*. These mechanisms are different in different languages.

For NLP it can be very advantageous if a natural language can be described formally like programming languages. The description of a programming language is generally given in the form of a context free grammar (*eg.* the grammar of the C language given in [27]). The *keywords* of the language and tokens such as arithmetic and logical operators, comma, semicolon, brackets, *etc.* are of pre-specified spellings. Variables, functions, *etc.* are referred by programmer chosen names, often called *identifiers*. Constants are also chosen by the programmer. When a program is written in that language to solve a particular real-world problem, then identifiers and constants chosen by the programmer are used to denote specific computational objects (values or logic). We can consider all the fixed spelling tokens in the grammar of a programming language at the same level as *fundamental tokens*. (Names of data types in the language may be treated as user chosen names.) Thus the context free grammar description of the programming language essentially describes how these fundamental tokens can be used together with identifiers to create programs. The grammar often makes distinction between different identifiers based on their declaration and

usage, such as function names, variable names, type names, *etc.* Further the grammar also specifies the implications of different portions of the program in the form of semantic description associated with the rules. The semantics is largely determined by the fundamental tokens.

Natural language expressions carry the intended meaning through the choice of the words therein and the sequence in which these words are used. In addition, in speech the meaning of an expression is significantly affected by elements such as pause, tone, *etc.* In written form some of these elements are roughly mapped to punctuation marks. The semantic roles of different words in a language are not same. There are the different categories of words such as *nouns, verbs, adjectives, prepositions, conjunctions, etc.* While some words have independent meanings, others do not. Let us, for the time being, call the former as type A words and the latter type B words. Nouns, verbs and adjectives are examples of type A words. Type B words perform certain functions in expressions in collaboration with other words, and are often referred to as *function words.* For example, the prepositions in English do not independently describe any object or concept from the domain of information that linguistic expressions seek to express. However, to make up an expression, prepositions play certain very important roles. We notice that in a given language the set of type B words is finite. It is the set of type A words that is not finite, and can grow as the domain of information to be encoded in the language expands.

With the above observations we consider that *a natural language is a framework for representing/encoding information and the type B words are an inherent part of this framework.* Type B words are part of a language's description, whereas type A words denotes points in the information domain and are not part of the basic description of the language. The (syntactic) description of a language refers to "categories of type A words", such as noun, verb, *etc.*, instead of individual instances of such words. Type B words in natural languages are like the keywords of a programming language, and type A words are like the identifiers or constants that denote specific objects or concepts in the domain of information to be represented by natural language expressions.

For a proper description of the syntax of a language it is necessary to know

what are the type B words in the language. To this end, we observe that type B words reflect certain universal characteristics of information that need to be reflected when the information is represented using a language. These characteristics may be-

- temporal aspects of events or existence (the concept of *tense*),
- specific *vs* non-specific instance of object or concept, (the concept of *determiner*),
- relative positions and directions (the concept of *sides*),
- action and cause-and-affect,
- relationships amongst objects and concepts, such as *owner, agent, instrument, etc.*,
- attaching property/attributes/qualities with objects (the concept of adjectives/qualifiers),
- degree of properties (concept of *quantification and comparison*),
- conditionality (concept of *if, else, but*),
- multiplicity (concept of *lists and conjunctions*),
- count (concept of *numbers*),
- inquisition,
- affirmation and negation,

  etc.

The above characteristics are reflected not only through complete words, but also through other elements of expressions such as affixes. In fact, this depends on the language. Syntactic structures, morphological transformations, phonological variations (stresses and pauses), *etc.* are different linguistic devices for encoding the above characteristics of information. The exact mix of these devises vary

across languages. For instance, the effect of prepositions in English is mostly realised by suffixes (case markers) in Assamese. This is so in several European languages as well ([41], *pp* 30). Even in English there are several suffixes. In the written form of a language, some punctuation symbols are also used to reflect some of the above characteristics. Hence we can extend our notion of type B words and term each of them as a *fundamental element of expressions* (referred to as *FEE* hereafter). We can visualise each FEE performing some specific *linguistic function. The set of FEEs is finite for a language.* The versatility of a language depends on the set of the FEEs and the gamut of functions they perform. But there is probably little difference in the versatility of different languages. Though the size of the set of FEEs in different languages might be unequal the range of functions that can be realised through them is very nearly the same. This hardly comes as a surprise, since the kind of characteristics of information (*not the information itself*) that each language is required to cater to is universal.

The other kind of words, *i.e.*, type A words, used in a language depends on the domain of information that is to be represented by linguistic expressions. In our scheme of analysis, these words play the role of *parameters* to the linguistic functions that the FEEs perform. Hence we term them as *parametric words* (referred to as PW hereafter). Note that we use the term PW to refer to the base form of such words. *This set of words is infinite.* Here we make the important statement that *acquisition of the FEEs of a language and their usage is a part of the core of the language acquisition.*

Expressions in a language are built by *appropriately* combining FEEs and PWs. The appropriateness is at three levels– (*i*) appropriate attachment of affixes to words (affixes can attach to FEEs too) and combining of multiple words (to get *compounds*), (*ii*) appropriate ordering of individual words, and (*iii*) semantic appropriateness. The first is morphology, the second is syntax and the third is semantics.

2.4.1 Morphology and syntax

A natural language expression represents some information from some domain. Hence there must be some PW in an expression. We also come across sentences

without any PW. For example

"Have you done the homework?"
"Yes".

Here the second sentence does not contain any PW. However, in the absence of the preceding sentence, the information conveyed by the second sentence would be expressed as "I have done the homework". Without the query in the first sentence, the second sentence does not convey any useful meaning. That is, in a discourse sentences often take abridged forms, existing with the tacit support of surrounding sentences. For the moment let us consider the unabridged expressions. The first sentence of the above example contains the following references from the information domain- *you, do, homework.* The characteristics conveyed are *inquisition, past perfect tense* (of *do*) and *specific instance* (of *homework*).

Meaning of a natural language expression beyond the direct meaning of the PWs is realised by combination of-

- use of FEEs attached to words, *eg.*, do $\rightarrow$ *done*,
- use of FEEs as distinct words themselves, *eg., the* homework,
- no explicit FEE, *eg.*, in *ripe mango* there is a relationship between the adjective *ripe* before the noun *mango* (both are PWs). In such cases an equivalent sub-expression using suitable FEEs can be visualised- *mango that is ripe.*

In other words certain portion of the entire meaning of an expression is realised through the FEEs. Attaching FEEs to words is morphology, while the other two are governed by *syntax* of the language. In languages where there is more use of FEEs as attachments to words, *i.e.*, the morphology is rich, the syntax details can be small. Where not many FEEs are attached to words, the syntax must fulfil the requirements. In fact, we observe that morphology arise out of highly regular occurrence of FEEs in strict order (as against free order) around PWs.

Morphology of a language can be looked at from two perspectives- the functions that the morphological features perform, and the exact form of the

morphological features. Morphological features originate from the requirement to achieve some function, *eg.* a suffix $-s$ in English originated to denote the *number* of an object. On the other hand, the form of the suffix may undergo slight variations (such as $-es$) depending on the word that it attaches to. Phonology can sometimes determine whether some pieces of morphological material are combinable at all ([7]). Again, in some cases the same function may be realised by more than one distinct morphological features. For example, in Assamese, plural of an object may be indicated by use of different suffixes such as *bor, khini, bilAk, samUH, etc.* (বোৰ, খিনি, বিলাক, সমূহ).

Work done in eighties regarding lexicon led to the realization that morphology is an autonomous module at par with the phonological and syntactic modules. On the other hand, syntactic systems capable of handling word formation operations in a more restricted way were developed during that period. Such systems could avoid many of the shortcomings encountered in earlier such efforts ([4]). Leiber states that in the conceptually simplest theory, all morphology would be part of the theory of syntax ([28]). However, most researchers have come to the conclusion that describing morphology within syntax is impossible and probably undesirable too. Rewrite schema and hierarchical structures proposed for morphology are systematically incompatible with notions of phrase structure and the tree structure proposed for syntax ([4]). Chomsky too, points out that syntax has properties completely unrelated to morphology, phonology and semantics ([41],pp 15).

2.4.2 Semantics

How much semantics is actually part of a language depends on our notion of language. Ideally the meanings of individual words in an expression and the structure of the expression together determine the meaning of the expression. However, in practice the meaning of an expression also depends on the discourse, pragmatics, and relevant world knowledge, much of which cannot be governed by any linguistic description. Often a given expression can be mapped to more than one meaning and it is not possible to resolve this ambiguity only through linguistic norms (*eg.*, [22]). Though it might be useful to have language govern

the formation of meaningful sentences, it cannot be enforced always. Because "meaningfulness" often depends on the domain and unless knowledge of the domain is sufficiently accumulated in the system, meaning of expressions cannot be verified. Hence it is safer to consider language as giving a structural framework for formation of expressions. Domain knowledge as much is available should be considered as an additional resource, which might help verify the consistency of the meaning figured for a given expression.

2.5 Language model for a computer

Like a human being, as an information processing entity a computer needs to be capable of using some language. Such linguistic capability must be achieved through appropriate computational power and not simply from principles of electronics that the hardware is based upon. The most basic capability of a modern digital computer is storage and recognition of a finite number of distinct physical quantities such as voltage, magnetic polarisation, etc., and performing arithmetic and logical operations involving these physical quantities. Using theories of computation and information representation, this capability is utilised for all kinds of information processing that we see.

The computations required for making a computer capable of dealing with a language depends on the language. We can take a look at programming languages to get some insight into computational aspects of language processing. Programming languages are consciously and carefully designed *by experts* so that they can be comprehensively processed by software created for the purpose. Nevertheless, in general it is one of the concerns to make a programming language as *natural* as possible. A programming language is described in terms of the syntax, *i.e.*, the structure of expressions in that language, and semantics, which is directly mapped to syntactic constructs. The syntax description comprises usage patterns of the various keywords, operators and some other symbols. Implementation of such a language essentially means implementation of computations that can translate information from such a language to some representation that the computer can directly deal with. These languages

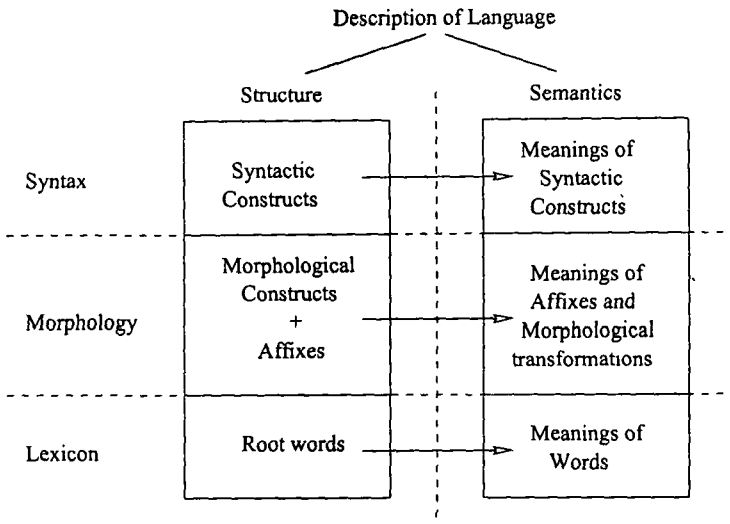


Figure 2.1: Description of a Language

facilitate convenient information representation for the programmer as well as for the computer. Natural languages, on the other hand, have evolved over long periods for conveying information between humans. Linguists hardly influence the evolution of the basic structure of a natural language. Some of the factors leading to the evolution of natural languages are, the need to cover larger and newer domains of information, convenience of usage, influence of other languages, need to enhance the appeal of expressions that can be formed, *etc.* The ease of processing a language by a computer is not a factor in this evolution. Programming languages are generally unambiguous, while ambiguity is common in natural languages. Still the model of programming language can be considered as a basis for computational modelling of natural languages – first, describe the structural details of the language, and then the semantic implications. Different languages may vary not only in the structure, but also in the *type of structure* – some may have elaborate rules of syntax while others may be richer in morphology. Figure 2.1 depicts this scheme of linguistic knowledge. Unlike the semantic aspects of expressions, the structural features can be observed without any other aid (source of information).

Possessing linguistic knowledge means being capable of mapping between "expressions" and their "meanings". Meaning of an expression is its information content. For such mapping to take place there must be some suitable format for representation of the information that an expression may carry. The information

representation format must be suitable for the kind of use that the information may be put to, and also for storage of the information. An expression itself is a representation of information and as such a natural language defines the format of this representation ([25]). This format is suitable for communication and storage purposes. For other uses of the information, other formats become necessary. However, there are several tasks involving linguistic competence where the goal is not as involved as mapping between expressions and meanings. For example, spelling checking, word sense disambiguation, syntax checking, *etc.* Many such tasks require structural knowledge of the language. In fact, such tasks are essential steps in the overall process of mapping between expressions and meanings. As shown in Figure 2.1, the structural description of a language comprises the set of words in the root forms, the set of non-word morphemes (*i.e.*, affixes) and their usage rules, other morphological transformations rules, set of categories of words, and the syntax rules in terms of the categories of words. This structural description of a natural language is generally large and can be incorporated in a computer only through careful planning.

2.6 Summary

In this chapter we have considered existing views on language acquisition. In line with our objective of computational acquisition of language, we have considered the general nature of human languages and their evolutionary development. We see that certain structural features of languages including the morphological features, are fundamental in nature, and their acquisition is essential for the acquisition of the language. For morphologically rich languages, acquisition of morphology is more important since morphology covers a large portion of the overall structural description of the language.

Chapter 3

The Assamese Language

The Assamese language belongs to the *Indic* branch (derived from Sanskrit) of the Indo-European family of languages. It is the easternmost New Indo-Aryan language used by about 15 million people in the state of Assam in north eastern India and its adjoining region (http://www.assam.org, http://www.assamcompany.org, http://www.assam.faithweb.com). There is a large volume of literary work and a rich ethnic culture based on the Assamese language. In this chapter we discuss mainly those aspects of the language that have a bearing on the problem of acquisition of the morphology and vocabulary, especially, in the unsupervised approach that we have adopted in our work.

3.1 Script

Assamese is written using the Assamese script, which is one of the several *Indic* scripts. It comprises 11 vowels, 34 consonants, and 10 digits. In addition, there are 7 symbols corresponding to certain consonant sounds, and all but the first vowel 'a', has a corresponding *operator* symbol to use with consonants. There are no *upper case* or *lower case* in Assamese script. The punctuation symbols used are same as those used in English except for the period. In Assamese a vertical dash instead of the dot, is used to imply the end of a sentence. The Assamese alphabet is same as the one used for Bangla and Manipuri (two other Indian languages) except for one consonant that is different in Assamese, and another

consonant that is found in Assamese but not in Bangla. Further, frequently *ligatures* (multiple consonant symbols combined into composite forms) called *juktakshars* are used to represent sounds that are actually combinations of the corresponding consonants.

The Assamese script is syllabic. The vowel 'a' is inherent with all consonants. The manifestation of this implied vowel is irregular. To produce the effect of the other vowels, the corresponding vowel operators are used. In contemporary spoken form of the language, speakers make little distinction between the following-

- Hraswa-i (ই) and dirgha-i (ঈ), and their corresponding operators (ি and ী).
- Hraswa-u (উ) and dirgha-u (ঊ), and their corresponding operators (ু and ূ).
- Pratham-sa (চ) and dwitiya-sa (ছ),
  murdhanya-ta (ট) and dantya-ta (ত),
  murdhanya-tha (ঠ) and dantya-tha (থ),
  murdhanya-da (ড) and dantya-da (দ),
  murdhanya-dha (ঢ) and dantya-dha (ধ),
  murdhanya-na (ণ) and dantya-na (ন),
  talabya-sa (শ), murdhanya-sa (ষ) and dantya-sa (স), and
  ri-kar and *ra-kar+hraswa-i-kar*, *eg.*, prithak (পৃথক) and pritam (প্রিতম).

Though the pronunciation of Assamese words generally can be directly figured from its spelling, irregularities such as the following exist-

- The implicit vowel 'a' at the end of a word- *eg.* মাছ is pronounced as *mAs*, whereas কাছ is pronounced as *kAsa*.
- Use of ligatures in certain situations- ligature is used to write *kalpanA* (কল্পনা) whereas no ligature is used to write *kalgas* (কলগছ).

3.2 Grammar

The history and philology of the Assamese language was scientifically studied and presented for the first time by Dr. Banikanta Kakati in 1935 in his doctoral

thesis ([26]).

Assamese grammar is described in several books on the topic. The first Assamese grammar, *A Grammar of the Assamese Language* by William Robinson was published in 1839. In 1848, N. Brown published an Assamese Grammar, and in 1894, Prof. Nicholl published his *Sketch of Assamese Grammar* [26]. In modern times the more comprehensive work on Assamese grammar was by Kaliram Medhi ([30] is the 3rd edition of the work). This and later published works such as [3], [20] and [32] are more commonly used today. The basic structure of Assamese expressions has similarity with that of expressions in other Indic languages.

Assamese is a *free word order* language to a large extent. For example, the sentence–

I shall go to school today.

can be written in Assamese in any one of the following forms-

মই আজি বিদ্যালয়লৈ যাম।
মই বিদ্যালয়লৈ আজি যাম।
মই যাম আজি বিদ্যালয়লৈ।
আজি মই বিদ্যালয়লৈ যাম।
আজি যাম মই বিদ্যালয়লৈ।
etc.,

where, the rough translations of the words are - মই (mai) = I, বিদ্যালয় (bidJAlay) = school, আজি (Azi) = today, যাম (jAm) = shall go, and suffix -লৈ (lE) = to. Some of the orderings have certain effect on the meaning conveyed. But the fact remains that all the above orderings are grammatically correct translations of the given English sentence.

Determiners : Assamese has a very rich set of determiners unlike most other Indic languages (see section on *Morphology* below). The usages of these determiners have some interesting aspects. Primarily the determiners are used as suffixes to nouns and pronouns according to certain subtle linguistic norms. For example, *mAnuhTo, phulkhini, gczopA, etc.* In many situations, the corresponding noun itself is not there, and a general pronoun plays that role. *eg. eiTo, eizn, etc.* In texts, determiners can also occur as separate words detached

from the object. For example, we can write *"l'rA To"* (ল'ৰা টো) or *"l'rATo"* (ল'ৰাটো) to mean "the boy", and *"chowAlI zanI"* (ছোৱালী জনী) or *"chowAlIzanI"* (ছোৱালীজনী), to mean "the girl".

A very important role of determiners is to indicate *number*. There are certain determiners that make the objects plural. For example, *bor, khini, brinda, bilAk,* (বোৰ, খিনি, বৃন্দ, বিলাক) *etc.* It is useful to compare and contrast the role of determiners of Assamese with that in other languages. In Hindi there is no morpheme corresponding to the basic (for singular number) determiners in Assamese, but plurality is achieved by certain affixations. For example, "boy", "the boy" and "the boys" in English are written in Hindi as "larkA", "larkA" and "larke". In Assamese, these would be "l'rA", "l'rATo", and "l'rAbor" (ল'ৰা, ল'ৰাটো, ল'ৰাবোৰ). In Bengali, these would be "chele", "cheleTA" and "chele gulo" (ছেলে, ছেলেটা, ছেলে গুলো). Though, in Bengali the use of determiners is similar to that in Assamese, but the number of different determiners for different types of objects is not as large as in Assamese.

Pronouns for second and third persons : Second person is referred to in three different categories using different pronouns-

- *tai* (তই) for junior or very close second persons,
- *tumi* (তুমি) for peers or slightly formal second persons,
- *Apuni* (আপুনি) for senior or formal second persons.

Similarly, the pronouns for referring to third person also are different-

- *si* (সি) for masculine and *tAi* (তাই) for feminine, junior or very close third persons,
- *teo** (তেওঁ) for peers or informal senior third persons,
- *tekhet* (তেখেত) for formal senior third persons,
- *terA* (তেৰা) for religious, revered third persons.

Foreign words : Like other Indian languages Assamese texts commonly contain foreign words, phrases, and abbreviations. Most of these are from English and some other Indian languages. Sometimes these are written in the original spelling (*i.e.*, using the foreign alphabet) and sometimes transliterated into Assamese script. Often such words are also subject to inflection. These facts put up a significant challenge in the task of creation of a computational lexicon.

Another type of words found in Indic languages is *nonsense words.* A nonsense word simply rhymes with the preceding actual word and roughly means *and such.* For example, *kitAp citAp* (কিতাপ চিতাপ), where *kitAp* means book and *citAp* simply imply "other things like book". Some nonsense words have some meaning in some other contexts. For example, *colA tolA* (চোলা তোলা), where *colA* means shirt and *tolA* can mean "pick" (imperative) in other contexts.

Verbs in Assamese : Compared to English there are fewer verbs in Assamese. In English many noun word-forms are also used as verbs to indicate action involving that noun. For example, shop, fish, present, dream, *etc.* In Assamese, corresponding effects, *i.e..*, the action related to a noun, are often achieved by use of some auxiliary verbs. *eg.*, *bazAr karA* (বজাৰ কৰা), *mAc marA* (মাচ মৰা), *pradAn karA* (প্রদান কৰা), *sapon dekhA* (সপোন দেখা), *etc.* In all these pairs the first word is a noun, and the second one is an auxiliary verb, and together each pair imply a particular action related to the noun. In such pairs the noun is generally not inflected. Some words such as *khelA*, however, are used as a noun as well as a verb. There is another kind of common ambiguity of syntactic roles – the verbs in second person imperative informal form are also used as adjectives derived from the verbs. *eg. khowA, diyA, etc.*

Negative of verbs : The negative sense of a verb in Assamese is obtained by use of a prefix with the verb. There is a class of prefixes for this purpose (see section on Morphology below). In Hindi this effect is achieved by use of the word "nahI*" before the verb. In Bengali, this is generally achieved by the use of the "-nA" and "-ni" (-না, -নি) suffixes.

Ambiguity of words : Ambiguity of syntactic category or sense of *commonly used words* is less in Assamese compared to English.

A major case of ambiguity in Assamese is the common use of valid words as proper nouns, viz., names of persons, institutions, *etc.*

3.2.1 Morphology

A discussion on Assamese morphology is presented in [26]. Assamese is a morphologically rich language. The morphology is largely concatenative. Secondary forms of words are frequently obtained by merging of root words and by affixation. Table B.1 shows over 550 different forms of the noun *l'rA* (ল'ৰা meaning *boy*), obtained by suffixation. Similarly, table B.2 shows over 500 forms of the verb *bH* (বহা meaning *sit*). The merging of words to obtain *compounds* is similar to those of other Indic languages to a large extent and generally follows the *sandhi* and *samas* ([47]) framework. For example,

AshA + atIt = AshAtIt (আশা + অতীত = আশাতীত)
kRhSNa + arjun = kRhSNArjun (কৃষ্ণ+ অর্জুণ = কৃষ্ণার্জুণ).

Inflection and derivation are achieved through affixation, *i.e.*, use of prefixes and suffixes. Use of suffixes in Assamese is more common than use of prefixes, and it is more extensive than in other Indic languages and English. A preliminary study showed that about 48% words in an Assamese text of around 1600 words are inflectional or derivational whereas only about 19% words in an English text of about 1400 words are so. Similarly, in a sample Hindi text of about 1000 words, 26% were inflectional and derivational.

In Assamese suffixes are used with verbs to convey tense, person and *parity* (viz, agreement with second person pronouns *toi* for junior or very close second persons, *tumi* for peers or slightly formal second persons, and *Apuni* for senior or formal second persons). Suffixes with nouns and pronouns convey case, number, *etc.* The suffixes indicating case in Assamese are called *bibhaktis* ([3, 20, 40, 30]). Assamese *bibhakti* are distinct from *bibhaktis* of other Indic languages. Suffixes also go with other words such as adjectives and adverbs. Apart from the inflectional suffixes there are several derivational suffixes too. In a large number

of cases of application of suffixes, spelling changes of the constituents do not occur.

A class of common suffixes in Assamese is that of the *determiners*. These are - *To, khn, khïni, zn, grAkI, bor, bilAk, zopA, znI, phAl, dAl, gAl, etc.* Such large number of determiners are not seen in other Indian languages and sometimes cause difficulty to non-native speakers of Assamese.

A very significant feature of Assamese morphology is the occurrence of *sequences of suffixes* in a single word. For example, *l'rAkeiTAkeino* = *l'rA* + *keiTA* + *k* + *ei* + *no* (ল'ৰাকেইটাকেইনো). The frequent occurrence of such sequences and the large number of suffixes in some of these sequences is a phenomenon that distinguishes Assamese from most other languages. In some cases Assamese also allows certain suffixes to be detached from the base part.

Assamese inherits 20 prefixes (*upasargas*) from Sanskrit ([3], [30]). There are additional prefixes specific to Assamese. Most prefixes in Assamese are irregular in the sense that they cannot be applied to a class of words in general. In many cases, prefixes change the meaning of words in such a way that the derived words may be treated as root words for the purpose of including in a lexicon. There are, however, few prefixes that indicates negation, number, *etc.*, whose use can be generalised for certain classes of words.

Prefixes for negatives of verbs : Negative forms of verbs are obtained by using one of the following prefixes with the verb- *na, nA, ni, nu, ne*, and *no* (ন, না, নি, নু, নে, নো). In most cases the prefix for a given verb is selected such that the vowel in the prefix is similar to the first vowel of the verb. For example,

na + Hay = naHay	(ন + হয় = নহয়)	means *is not*
nA + lAge = nAlAge	(না + লাগে = নালাগে)	means *not needed*
ni + dio* = nidio*	(নি + দিওঁ = নিদিওঁ)	means *(I) will not give*
nu + buzA = nubuzA	(নু + বুজা = নুবুজা)	means *(you) do not understand*
ne + dekhA = nedekhA	(নে + দেখা = নেদেখা)	means *(you) do not see*
no + khole = nokhole	(নো + খোলে = নোখোলে)	means *(it) does not open.*

However, the prefix *ne* can also be used with verbs with *A* as the first vowel. For example, *nezAo*, nekhAy, etc.* (নেজাওঁ, নেখায়). There is another similar prefix *nO*

(নৌ) which means *(even) before.* For example,

nO + pao*tei নৌ + পাওঁতেই which means *before (one) got (it).*

Prefixes and quantification : There are certain nouns which mean some object and they also represent some quantity. This is true in Assamese as well as many other languages. For example, day, bucket, glass, *etc.* In Assamese there are many more such nouns. such as *ghar* (ঘৰ), *buku* (বুকু), *k*kAl* (কঁকাল), which mean *house, chest, waist* respectively. In English these nouns can be associated with quantities by transformations such as *houseful, chestful, waist-deep.* Now, before such nouns a number can be placed to multiply the quantity that is represented. For example, *cAri din, dui bAlti, tini gilAc* which mean *four days, two buckets, and three glasses* respectively. In Assamese the numbers one (এক ek), two (দুই dui), and six (ছয় Cay) in such situations are often attached to the noun as prefixes For example, এবুকু, দুদিন, ছমাহ read as *ebuku, dudin* and *CmAH* mean *one chestful (esp. love), two days (duration)* and *six months (duration),* respectively.

To denote the number of some countable noun (object), the number is placed before the determiner to be used with the noun followed by the noun itself. The determiner in these must not be ones that denote plurals. Also, the determiner *TA* (টা) is used with the *number* instead of the regular determiner *To* (টো). For example,

cAri+TA l'rA = cAriTA l'rA (চাৰি+টা ল'ৰা = চাৰিটা ল'ৰা == *four boys*)
ek+khan desh = ekhan desh (এক+খন দেশ = এখন দেশ == *one country*)
du+zanI gAy = duzanI gAy (দু+জনী গায় = দুজনী গায় == *two cows*).

In the above examples, note that the determiners that may be normally used with *l'rA* (ল'ৰা == boy), *desh* (দেশ == country) and *gAy* (গায় == cow) are *To* (টো), *khan* (খন), and *zanI* (জনী) respectively.

3.3 Lexicon

There is no existing computational lexicon of Assamese. There are several linguistic dictionaries of the language, viz., *Hemkosh, Chandrakanta Abhidhan* ([2, 32]), *etc.* The first Assamese-English Dictionary was compiled by M. Bronson

and published by the American Baptist Missionaries in 1867 [26]. *Hemkosh* was originally published in 1900 and had over 22000 entries. *Chandrakanta Abhidhan* was first published in 1933 and had about 37000 entries. The later editions of both were enlarged.

3.4 Encoding of Assamese text in computer

Scripts of natural language are visual patterns on some surface. In a computer these natural language texts may be captured as *images*. But, for convenience of analysis, manipulation and storage, a convention of representing such texts as a sequence of numeric codes is followed. Each elementary symbol of the script is represented by a unique numeric code. Hence the entire text, which is a sequence of the elementary symbols of the script, can be represented by the sequence of numeric codes. To reproduce the text internally represented inside a computer using numeric codes for human use, suitable glyphs (fonts) are produced for each numeric code value. In practice font sets are defined such that each individual font in the set can be addressed by a number. For Roman script, the ASCII encoding scheme is used and each numeric code uniquely identifies the font corresponding to it. To cater to other script systems the Unicode encoding system has been defined. However, the visual glyphs for each numeric code value are not part of the Unicode. For languages such as Assamese, reproducing the text for reading is somewhat complex since the same internal code may have to be displayed using different glyphs in different contexts. Also, the glyphs corresponding to two sequentially occurring numeric codes may require to be displayed in formations other than simply one followed by the other. Since the Unicode does not enforce a standard for the fonts, different font sets are in use which effectively imply different encoding schemes. These may be called *font encodings*. Presently, font encodings such as *Aadarsha Ratneswar, Luit, Kamakhya, Ramdhenu, etc.* are in use for Assamese texts. Any software that needs to analyse texts in these various encodings may have to first carry out transliteration of the text into a standard encoding. It is also possible to have a script system in which each Assamese letter is denoted by a distinct Roman letter or letter sequence so chosen that the text

can be read out directly. Such a Roman transliteration scheme has been used in our work as a standard encoding for Assamese texts. It is described in Appendix A

3.5 Summary

In this chapter we have briefly presented the salient features of the Assamese language, that on one hand, highlight the richness and prominence of morphology of the language, and on the other, indicate the kinds of issues that can arise in the computational acquisition of morphology from a text corpus.

Chapter 4

Identification of Suffixes from a Text Corpus

4.1 Introduction

An NLP system designed to understand the expressions of a language must be able to recognize the words in the expressions and know the meanings of those words. The set of all words that may occur in expressions in a given natural language is very large and a practical system has knowledge of only a subset of these words. This subset is the *vocabulary* of that system. In addition, it is desirable that a system gracefully deals with words outside its vocabulary. Recognizing a word implies determining various attributes of the word. A word of a language has several attributes such as part-of-speech (POS), tense (if the POS is verb), number, *etc.* Meaning of a word is another attribute of the word, which maps the lexical form of the word to a real world object or concept. A word is only partially recognized if some, but not all, of its attributes are determined. Often during analysis of an expression, even partial recognition of some words is useful in understanding the context. Certain linguistic features, the context of the expression and knowledge of the environment often make it possible for a system to interpret unseen words. Context is realised through analysis of the morphology, syntax and meaning of the *recognized* portion of the expression. Relevant knowledge of the environment, in turn, must be selected

based on the context. Hence both these factors require analysis of meanings of the recognized portion of the expressions. By *linguistic features* we mean certain language dependent surface level patterns, which can help guess attributes of unseen words. These patterns may be collocations, word structure (such as affixation), syntactic analogy, *etc.* Of these, word structure, *i.e.*, morphology, is very significant in certain languages.

In this chapter we develop an unsupervised approach to identify the morphological features of a highly inflectional language, from a text corpus. More specifically, we try to identify the suffixes and their usage patterns, since suffixation is the predominant morphological phenomenon in Assamese, the language that we are focusing on. We have considered existing methods, and identified ways in which more effective performance can be achieved. The major deliverables of the exercise described in this chapter are a set of suffixes and suffix-sequences, and a morphological lexicon.

4.2 Incorporating morphological knowledge into an NLP system

Morphology determines how each word in an expression is built from component parts. A predominant form of morphology is the *concatenative morphology* where a word form is obtained by combining two or more *morphemes*. There are other morphological transformations where a word is modified without combining additional morphemes to it. Whatever is the type of transformation, for an NLP system morphology as a phenomenon is significant because–

- without a proper computation of morphology, all word forms that occur in expressions need to be recognized individually, and,

- morphology brings out the attributes of words that help in recognizing the structure of a sentence.

Recognizing all word forms is likely to impose a huge demand on a system, and this requirement alone may make building NLP systems prohibitive. On the other hand, the morphological knowledge needed to understand the structure of the individual words used in a language is generally finite and can be quite effectively

encoded in a computational system. Computational encoding of morphological knowledge enhances the capability of the system to recognize various word forms. Another fact that makes morphology significant is that each transformation performs some clear linguistic function; thus, recognizing the morphological features in a word implies recognition of some part of its meaning. In other words, morphology by itself leads to a partial recognition of a word.

To encode the morphological phenomena of a language in an NLP system, each phenomenon must first be identified. Depending on the nature of a phenomenon, it must be implemented in suitable computational terms. For the identification of the morphological phenomena in a language, one must study relevant literature published by linguists. However, often either the results of linguistic studies are not readily available, or the format of the available information is not suitable for computational purposes. Thus, approaches used by morphological processing systems range from hand-coding of morphological "rules" provided by linguists in software, to automatic identification of morphological rules from examples of text inputs. Perhaps the most widely cited work on hand-coding morphological rules is the Porter's method for stemming ([36]). This method deals with suffixation morphology. Others have subsequently attempted to improve this method (*eg.* [39]).

Automatic identification of morphological rules generally analyses an input corpus in order to discover the underlying morphological features. These approaches are, in turn, of two broad types– one, the input corpus is annotated, and two, the input corpus is raw, *i.e.*, unannotated. For instance, Daelemans takes as input a POS tagged corpus for the lexical acquisition task [15]. Approaches that take unannotated corpus as input are unsupervised approaches. Most unsupervised approaches are primarily probabilistic (*eg.*, [19, 12, 13, 14]). On the other hand, the unsupervised approaches such as the one described in [16] are not exactly probabilistic. In our method too, we rely on some "string-matching", statistical support and set-theoretic principles for the task.

Speech is the primary form of expression for natural languages. The evolution of linguistic features including morphology, is based on this spoken form. Hence, one important approach to acquisition of morphology is to consider

the phonological form of utterances. For example, Gasser [18] describes a connectionist approach that takes as input phones and outputs the associated roots and inflections.

4.3 Motivation

A child learns her (or his) native language by subconsciously analyzing sentences that she (or he) hears. She neither consults a dictionary nor gets any explicit instructions on grammar or vocabulary, but she can perceive the real-world entities and events in the environment that the sentences describe. That she can relate the sentences to real-world events and entities, is very important in this learning process, especially, at an early age. For example, suppose the child hears the sentences "There is a cat under the chair", and "The cat is brown" describing two separate situations that she sees. Because of the presence of the particular object "cat" in the two situations it becomes evident to the child that the word "cat" in the two sentences denotes that particular object. In this respect the natural language is actually an "artificial" scheme of representation of the real-world facts[1]. This scheme is acquired by the child by identifying the correspondence between elements of the natural language expressions and the elements of the real-world situations. Thus the perception of the environment itself serves as a representation of the information, essential for the language acquisition task. In the case of a computer processing a natural language expression provided as input, generally the computer does not possess any facility to gather relevant knowledge about the environment. Usually the system is unable to draw any relationship between natural language expressions and real world entities, events and concepts to facilitate or expedite the comprehension process. In a sense, the processing is limited to the "structure" only and does not involve its "meaning". Further, in a child there is an inherent urge to communicate information that is drawn from the environment as well as produced

[1]The real object "cat" is denoted by different artificial words in different languages, such as *cat* in English, *mekurI* (মেকুৰী) in Assamese, and so on. Similarly, the rules of syntax and semantics are artificial and hence they are different in different languages to describe the same real situation.

within the brain. Hence a child makes effort to acquire a language. In a computer, there is no inclination for communication other than that enforced by the software that is installed. However, computers are good at processing data. As far as apparent structure of expressions is concerned, if there is any regularity in the structure, a computer program should be able to extract it. In case of the structure of a linguistic expression, the structure due to morphology is highly regular. The method that we describe here attempts to build a lexicon and learn morphological rules of a language by studying texts of the language and without any direct manual specification of the language.

A motivation for the particular approach taken in our work for morphological analysis is that in Assamese formation of derivatives from the root or base forms of words is ubiquitous, the language being very inflectional. We believe that for further computational processing of Assamese text, it will be very useful to analyse words to identify the root form and the exact nature of derivation used in each case. To handle such a task, unsupervised learning of morphology is useful. Our algorithm uses techniques distinct from those described in [19].

4.4 Morphological phenomena in Assamese

As mentioned in Section 3.2.1, Assamese is a morphologically rich language. A large proportion of the words that occur in Assamese expressions are obtained through morphological transformations of base words. Assamese morphology is largely concatenative with three prominent operations, *viz.*, prefix, suffix and compound formation. These are sometimes accompanied by change in spellings of the root words, *eg.*,

$$par + adhIn = parAdhIn \text{ (পৰাধীন)}.$$

In this work, we do not target analysis of compounds. The other kind of morphological phenomenon, *i.e.*, *affixation* is differentiated into two types, *inflectional* and *derivational* (such as in [1]), *eg.*,

Inflectional : $mAnuH + e = mAnuHe$ (মানুহ, মানুহে)

Derivational : $drirha + tA = drirhatA$ (দৃঢ়, দৃঢ়তা, = *firm, firmness*).

In our exercises in acquisition of the morphology of Assamese, we treat them

alike. Further, we focus on suffixes since suffixes are more common in Assamese and embody significant linguistic information. In addition, we take into account the phenomenon of *sequence of suffixes* occurring in a single word.

4.5 Terms and notations used

A word comprises one or more *morphemes*. A decomposition of a word is a representation of the word as a sequence of two or more *parts*, where each part, in turn, comprises one or more morphemes or is empty. An empty part is written as $NULL$. For example the word *cheerfully* can be decomposed as

[cheerfully = cheer + ful + ly]
[cheerfully = cheerful + ly]
[cheerfully = cheer + fully]
[cheerfully = cheerfully + $NULL$].

With respect to the original letter sequence of a word, we refer to the position where two parts are separated as *morpheme boundary*. In our exercise of unsupervised analysis, words are sometimes erroneously broken up at points other than morpheme boundaries too. Hence, we refer to the computationally identified points of separation of two parts of a word, as *partition points*. If we keep out the cases of prefixes in words, then the first part in the sequence is the *base* of the decomposition. The portion of the sequence after the base is the *morphological extension* or simply, *extension*. The base has a distinct *independent* meaning and can generally occur as a word by itself. Sometimes the base indicated in a decomposition is not an independent word in that form. For example,

[computer = comput + er]
[computing = comput + ing]
[computation = comput + ation],

are decompositions where the independent word form of the base is *compute*. Alternatively, we may write the base in its independent form, implying that the

concatenation of the subsequent part is accompanied by spelling modification –

[computer = compute + er]

[computing = compute + ing]

[computation = compute + ation].

Examples of decompositions of Assamese words are

$[karA = kar + A]$ ([কৰা = কৰ + আ])

$[baHalkE = baHal + kE]$ ([বহলকৈ = বহল + কৈ])

($karA$ = do (*imperative*); $baHalkE$ = spread out / in detail (*adverb*)).

The apparent spelling modification in the decomposition [কৰা = কৰ + আ] is trivial, or *regular*, since a vowel (আ) after a consonant (ৰ) is usually represented by the corresponding vowel operator (া). In Assamese there are very few cases where the bases do not occur as independent words. For example,

$[p\_rshiXit = p\_rshiX + it]$ ([প্রশিক্ষিত = প্রশিক্ষ + ইত])

$[p\_rshiXk = p\_rshiX + k]$ ([প্রশিক্ষক = প্রশিক্ষ + ক])

($p\_rshiXit$ = train-ed (*adjective*); p_rshiXk = train-er)

where the independent word form of the base is p_rshiXN (প্রশিক্ষণ), meaning, *training (noun)*.

In general a decomposition can be specified by the 3-tuple

$$< w, b, x >,$$

where, w is the word being decomposed, b is the base and e is the morphological extension. The morphological extension e can be a single part, a sequence of parts, or NULL. We denote it in the more readable form

$$[w = b + x].$$

If the concatenation of the constituent parts of a decomposition does not require any spelling modification, or any spelling modification required is *regular*, the word itself need not be specified with the sequence of the constituent parts in the decomposition. We may denote such a decomposition as

$$[b + e].$$

The morphemes in the morphological extension are broadly of two types– *suffixes* and *compound parts*. A suffix is a morpheme that does not have an independent meaning of its own, and only modifies the meaning of the portion

preceding it in the word. For example, the morpheme *ly* in *cheer+ful+ly*. A compound part, on the other hand, is actually a word by itself with an independent meaning and in the given decomposition forms a compound with the base. For example, the morpheme *over* in the following decomposition is a compound part

$$[\text{switchover} = \text{switch} + \text{over}].$$

As a special case, we can decompose a word with itself as the base and an *empty* (or $NULL$) morphological extension. We refer to such a decomposition as a *trivial decomposition*. For example,

$$[\text{cheerfully} = \text{cheerfully} + NULL]$$

is a trivial decomposition. If for a word no non-trivial decomposition is possible, the word is a *root word*. On the other hand, when the morphological extension in a decomposition is non-null, the word is a *derived word* or a *derivative*. A decomposition is *valid* if linguistically (semantically) the derivative is formed from the base with the given morphological extension.

We use the following conventions to symbolically denote items such as letters, letter strings, morphemes, words, *etc.*

$m,\ n,\ o,\ q$	- letter
$a, b, ..., k, \alpha$	- letter string
μ	- morpheme
$w,\ \omega$	- word
β	- base
π	- root
σ, s, t, u, v, y, z	- suffix (single morpheme)
$p,\ \rho$	- part comprising one or more morphemes
x	- morphological extension
δ	- decomposition.

Note that in real examples the letters do not assume the above meanings.

Wherever required, we use subscripts with the symbols to distinguish amongst them. Use of subscripts also imply that the subscripted letters are being used in symbolic sense and not as real strings. A morpheme, word, base, root, suffix and a part are special cases of letter strings. To denote a sequence of such items we

use '+' between the adjacent items. A morphological extension is a sequence of one or more parts. A decomposition contains a parts-sequence, of which the first part is the base. To denote the concatenation of a sequence of items, we write them as a string without any separator in between. Further, $|X|$ denotes the length of X, where X is a sequence of one or more parts. Length here implies the number of letters in the non-NULL parts of the parts-sequence, plus, the number of '+' preceding non-NULL parts in the sequence.

We use the term *morphological expression* to refer meaningful constructs involving items mentioned above. The items in a morphological expression may be either symbolic or real. For example:

1. $w = \alpha$
2. $ful + ly$
3. [কৰাগৈ = কৰা + গৈ]
4. $w = \rho\sigma$
5. δ : $[w = p + x]$

   etc.

In real morphological expressions, the items can be either English or Assamese. Assamese items, in turn, may be either in Assamese or Roman script. Wherever necessary, to distinguish between different types of morphological expressions containing Roman letters, we attach the special subscript $\mathcal{S}$ to symbolic morphological expressions[2] and the special subscript $\mathcal{A}$ to Assamese morphological expressions in Roman letters. For example,

1. $[w = b + s]_{\mathcal{S}}$ 2. $karAgE_{\mathcal{A}}$ 3. $(karA + gE)_{\mathcal{A}}$.

When these subscripts are to be attached to an expression consisting of more than a single string of letters (*eg.*, a sequence of parts), the expression must be enclosed within brackets. While expressions denoting decompositions are enclosed within square brackets, other expressions are enclosed within round brackets for this purpose, like in the third example above. For an expression consisting of only a single string of letters, the brackets may be omitted (*eg.*, $karAgE_{\mathcal{A}}$). Further, the items of a subscripted expression can be individually referred to with the

[2]Special subscript $\mathcal{S}$ is required only if the symbolic items do not already have any other subscripts.

subscript attached. For example, we may refer to the first part of the sequence $(a+b)_s$ as a_s.

At times, immediately following an Assamese word or decomposition, we provide its meaning in English (in case of decompositions, the meaning is that of the word being decomposed, unless otherwise specified) enclosed between two (slash) characters. For example,

$[karA = kar + A]_A$ [করা = কৰ + আ] /do (*imperative*)/.

4.6 Acquisition of morphology from a text corpus

For acquiring morphology of a language from an unannotated text corpus we perform a *surface level analysis* of the corpus. An unannotated text corpus presents two kinds of information about the language– first, the lexical space of the language, *i.e.*, of the infinite possible letter sequences, the ones that form valid words in the language, and second, the morphological phenomena, *i.e.*, the noticeable similarity in the structure of groups of words. In case of Assamese the predominant morphological phenomenon is suffixation. We model the morphology acquired through analysis of the input training corpus, in the form of a collection of suffixes and the criteria for identifying the presence of these suffixes in different words. This knowledge of morphology is used in building a lexicon that is compact as well as provides more insight about words than a plain listing of the words encountered does. The morphological model and the lexicon can be subsequently used for morphological analysis of words in texts.

Our first task is to identify the underlying suffixes in the language. Suppose, S_C is the set of suffixes identified by the computational process, and S_L is the set of suffixes that are actually there in the language. The ideal goal of the morphology acquisition process is to have S_C be the same as S_L. However, due to the constraints on the available evidence and the methods applied, S_L is usually not the same as S_C. Letter strings that are not really suffixes are identified as ones, while several valid suffixes are left unidentified. A morphology acquisition method is useful only if the S_C obtained is a close approximation of

an underlying S_L. Similar issues arise in the next task of our approach. The next task is to build a lexicon from a corpus– possibly, the same training corpus. We use the suffixes acquired to decompose the words in the corpus. Simply looking for matching of the suffixes at the end portion of words leads to a large number of invalid decompositions. Methods discussed in [19, 16] are representative of reported approaches to tackle the problem. In our approach, we apply heuristics based on statistics as well as other language specific and script specific aspects. We find that in case of Assamese, and possibly in other languages with similar features, our approach produces better results. We first briefly discuss the two approaches proposed by Gaussier and Goldsmith for identification of suffixes from a text corpus.

4.6.1 Gaussier's approach

In [16] Gaussier presents a method to acquire the suffixes used in a raw text corpus. The main idea in this method is to find pairs of morphological extensions that have occurred with the same base, and then, for each pair of morphological extensions, to find at least one more base that has occurred with that pair. Finding pairs of morphological extensions for a base essentially strives to ensure that the base is regular and hence likely to be valid. Then, looking for at least one more base that has occurred with a pair of morphological extensions, essentially strives to ensure that the morphological extensions are regular and are likely to be true suffixes.

More specifically, a pair of decompositions using a common base β is obtained for words w_i and w_j in the input corpus such that-

$$[w_1 = \beta_1 + \alpha_1]$$
$$[w_2 = \beta_1 + \alpha_2],$$

where, the $|\beta_1| \geq p$, $|\alpha_1| \geq 0$, and $|\alpha_2| > 0$. If $|\alpha_1| = 0$ (*i.e.*, α_1 is $NULL$), β_1 must be a word in the input corpus. The language independent value of p suggested is 5. The morphological extensions, α_1 and α_2, together referred to as a *pseudo-suffix pair*, are accepted as valid, if for some words w_3 and w_4 in the input, the following decompositions hold

$$[w_3 = \beta_2 + \alpha_1]$$
$$[w_4 = \beta_2 + \alpha_2],$$

where, $\beta_1 \neq \beta_2$. That is, the pseudo-suffix pair is accepted if the pair applies to at least two distinct bases.

In general, the criteria involved in this method can be stated as

1. Minimum base length, $p = 5$.
2. Minimum morphological extension co-occurrence, $c = 2$, *i.e.*, two suffixes occurring with the same base are considered as a pair In other words, each base must have occurred with at least two morphological extensions[3].
3. Minimum morphological extension occurrence, $f = 2$, *i.e.*, each morphological extension must occur with at least two bases.

The experimental results obtained upon implementing Gaussier's method and testing on an Assamese corpus of about 1,16,000 words (**corpus A**) from 231 newspaper articles, are summarized in table 4.1 and shown graphically in figure 4.1.

Total number of distinct words · 20140
Actual number of suffixes present 187

Base length threshold, p	B	C	Q	S	Precision (%)	Recall (%)	*f-measure* (%)
3	463	398	323	160	25 68	85 56	39 51
4	203	206	244	149	42 33	79 68	55 29
5	94	67	149	120	56 07	64 17	59 85
6	34	27	110	90	72 58	48 13	57 88
7	15	5	73	54	78 26	28 88	42 19

S Suffix, Q Suffix-sequence, C Compound parts;
B· Invalid morphological extension

Table 4.1· Summary of results of method proposed by Gaussier, with different values for p

[3]One of the morphological extension in the pseudo-suffix pair can be NULL, which means that the base has occurred as an independent word

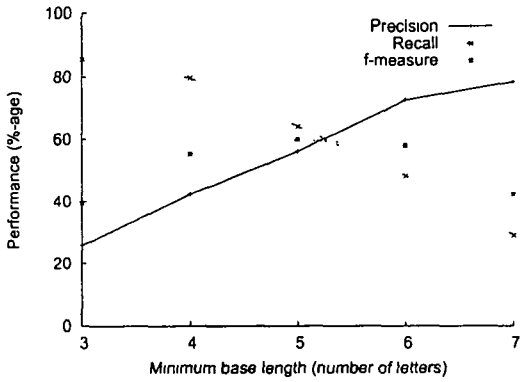


Figure 4.1: Effect of base length, p, in Gaussier's method

4.6.2 Goldsmith's approach

Some unsupervised morphology acquisition methods (*eg.* [44]) are based on probabilistic models. A particularly interesting approach, which can be seen as a special case of probabilistic idea is presented by Goldsmith ([19]). It is based on the Minimum Description Length (MDL) concept. The main intuition in this concept is that if all the morphemes, which are the basic elements of all words, involved in an input are assigned distinct numeric values in the smallest possible number space, the input can be represented as a sequence of these numbers. Identification of morphemes can be guided by the goal of minimizing the length of the representation of the input corpus, which depends on the total number of morphemes as well as the representation lengths of the individual morphemes in number of bits

The implementation of this approach is available as a free downloadable software called *Linguistica*. We test this software over the same corpus A as used for our own method as well as the method discussed in section 4.6.1. For this test the input corpus needs to be preprocessed since in the Roman script based encoding scheme used in the corpus, certain Assamese letters are represented using more than one Roman letter, and some are represented using non-alphabetic characters (see Appendix A) The results of this experiment are summarized in Table 4 2.

Number of input words	: 116096
Number of distinct input words (including hyphenated words)	: 20685
No. of distinct morphological extensions found, n	: 167
No. of distinct valid suffixes identified, s	: 80
No. of distinct suffixes that should be further broken up, q	: 57
No. of morphological extensions that are compounds parts, c	: 21
No. of invalid morphological extensions, b	: 9
Actual number of suffixes present in the input, S	: 187
Precision of single suffix idenfication $(s/(s+b))$	: 89.89%
Recall of suffix idenfication (s/S)	: 42.78%
Proportion of non-invalid morphological extensions to total morphological extensions $((s+q+c)/n)$	: 94.61%
f-measure	: 57.97%

Table 4.2: Summary of results from *Linguistica* (Goldsmith's method [19])

4.7 Our approach for morphology acquisition

To discover the set of suffixes in Assamese from a raw text corpus, our first step is somewhat similar to Gaussier's approach (see section 4.6.1). We obtain all the decompositions in each of which a word, w_1 of the corpus is obtained by appending a string of letters α to another word, w_2, of the corpus. That is

$$[w_1 = w_2 + \alpha] .$$

We refer to this exercise as *initial decomposition* (it is also used as a noun phrase to indicate the outcome of this exercise). The idea is that since w_2 occurs as a word in the corpus, it has an independent meaning. Hence, it can be the base for some decomposition. Since word w_1 has w_2 as its leading portion, it is likely that w_1 is derived from w_2 with α as a morphological extension. The method of Gaussier ([16]) detects a morphological extension, x_s, if at least two bases that occur with x_s, also occur with some other, possibly *NULL*, suffix y_s. *i.e.*, if the words ax_s, ay_s, bx_s and by_s occur in the corpus, we get the pseudo-suffix pair $(x,y)_s$. In our initial decomposition we miss a suffix x_s even if it occurs adequate

number of times unless the corresponding bases also occur independently. In Assamese this does not adversely affect the recall since given a good corpus size, bases do occur without the suffixes, too.

Suppose, the set of words in the input corpus is W. We find the set of initial decompositions, D, as

$$D : \{[w = b + x]_S \mid (w = bx)_S, \text{ and } w_S, b_S \in W\}.$$

The set of morphological extensions obtained is

$$E : \{x_S \mid [w = b + x]_S \in D\}.$$

A few sample decompositions are shown in Table 4.3. In general, it is observed that– from some bases more than one derivative are obtained by adding different morphological extensions, some bases are further decomposed using other bases, some derivatives remain undecomposed, some of the morphological extensions require further break up, and some of the decompositions found are actually invalid. Also, some morphological extensions contain compound parts (*eg.*, line 6 in Table 4.3).

1.	$[kitApar = kitAp + ar]_A$	(কিতাপৰ)	/of book(s)/
2.	$[kitApat = kitAp + at]_A$	(কিতাপত)	/in book(s)/
3.	$[kitAparHe = kitAp + arHe]_A$	(কিতাপৰহে)	/exactly of book(s)/
4.	$[kitAparHe = kitApar + He]_A$		
5.	$[kitApkhanar = kitAp + khanar]_A$	(কিতাপখনৰ)	/of the book/
6.	$[m/nt\_rIpd = m/nt\_rI + pd]_A$	(মন্ত্রী পদ)	/minister's post/
*7.	$[kalaH = kal + aH]_A$	(কলহ)	/pot/
			kal_A = banana.

The last decomposition marked * is invalid, since *kalaH* (কলহ) is actually a root word not related to *kal* (কল).

Table 4.3: Some sample decompositions

In Assamese texts sometimes the single-quote mark is used between two morphemes that are fused. This is generally seen with foreign words to which some suffix is added. For example,

$[HAiwe'r = HAiwe + r]_A$ [হাইৱে'ৰ = হাইৱে + ৰ] /of highway/ .

The single-quote used in the suffixed word does not play any role in the

pronunciation of the word, and might make $'r_{\mathcal{A}}$ appear as suffix instead of $r_{\mathcal{A}}$. We cannot remove this mark entirely from the list of input words since it is part of the spelling of the root word, such as

$$m'H_{\mathcal{A}} \quad (\text{ম'হ}) \quad /\text{mosquito}/ \ .$$

Hence, we remove the single-quote mark from the beginning and end of the morphological extensions. We perform this for the results of the other methods too.

Relevant statistics for finding the decompositions from corpus A of newspaper articles is given in Table 4.4. The exercise identifies almost all the suffixes (high *recall*, 99.11%), but along with too many "non-suffixes" (only 1.65% are suffixes). A morphological extension is either a suffix (*i.e.*, single morpheme), a composite suffix (*i.e.*, sequence of suffixes), a compound part possibly followed by one or more suffixes, or just invalid (none of the previous three cases). For calculating the recall, we have not referred to any pre-defined set of suffixes of the language. Instead, we have identified and referred to the set of suffixes that are *actually present* in the corpus. A list of suffixes in a larger corpus is given in Table 4.14.

4.8 Selecting valid suffixes from the initial decompositions

The set of morphological extensions E obtained from the initial decompositions has a recall close to 100%, but there are several non-suffixes too. Of the morphological extensions that are not suffixes, about 21% are either composite suffixes (suffix-sequences) or sequences with compound parts. The decompositions involving such morphological extensions are actually valid. Thus, about 58% of the initial decompositions are valid. In the other methods too, especially Gaussier's, such cases are there. Also, some invalid decompositions have valid morphological extensions, but the derived word and the base are not semantically related. For example, consider the decompositions

1. $[bizy = bi + zy]_{\mathcal{A}}$ [বিজয় = বি + জয়] /victory/
2. $[bi/shwzy = bi/shw + zy]_{\mathcal{A}}$ [বিশ্বজয় = বিশ্ব + জয়] /world-victory/.

Number of input words	: 116096
Number of distinct input words (original count was 20685, but in hyphenated words only the last components are retained).	: 20140
Number of decompositions (including multiple for same word).	: 29054
No. of distinct morphological extensions in the decompositions, n	: 13715
No of distinct valid suffixes identified, s	: 185
No. of distinct suffixes that should be further broken up, q	: 654
No. of morphological extensions that are compounds parts	: 2218
No of invalid morphological extensions	: 10658
Actual number of suffixes present in the input, S	: 187
No of distinct bases that occur in decompositions	: 5186
No. of bases that occur in more than one decompositions	: 2820
No of bases that are, in turn, decomposed, too	: 3638
No. of invalid decompositions	: 12234
Precision of single suffix idenfication (s/n)	: 1.35%
Recall of suffix idenfication (s/S)	: 98.93%
Proportion of non-invalid morphological extensions to total morphological extensions $((s+q+c)/n)$	: 22.29%

Table 4.4: Summary of initial decompositions from a corpus of 231 newspaper articles

The morphological extension $zy_{\text{অ}}$ is valid in the second decomposition, but not with the base $bi_{\text{অ}}$ in the first The string $bi_{\text{অ}}$ is not a valid word. It is actually a prefix, and it has probably occurred in the corpus as an initial in some abbreviated word, such as

$bizepi_{\text{অ}}$ (বিজেপি) /BJP/.

We take a sequence of measures to achieve better performance in selection of suffixes from the initial decompositions. First we try to reduce the number of invalid decompositions. Then we try to distinguish the suffixes from the other morphological extensions. We try to break up the composite suffixes to reveal the suffix constituents.

Precision, recall and f-measure of the initial decompositions:

The concept of decomposition, on which the method described so far is based, does not distinguish between single suffixes, composite suffixes and sequences with compound parts. There are a large number of valid decompositions involving composite suffixes (sequences) and compounds, the precision computed as the ratio of single suffixes to all the morphological extensions is much lower than 100%. Before we take steps to distinguish amongst the different valid morphological extensions, our immediate objective is to get rid of the invalid decompositions. Till then we compute the precision of the process as the ratio of the number of single suffixes to the sum of the number of single suffixes and the number of all invalid morphological extensions. We compute the recall of the exercise as the proportion of single suffixes identified to those actually present in the corpus. An aggregate of precision and recall is the *f-measure* ([8]). We compute it as–

$$f = \frac{2 * S * 100}{S + B + T}$$

where, S is the number of suffixes identified, B is the number of invalid morphological extensions, and T is the total number of suffixes present in the input.

4.8.1 Frequency of morphological extensions

A simple intuitive idea for selecting valid decompositions from the initial decompositions' set, D, is to look out for *regularity* of the parts in the decompositions. First we try to ensure that the morphological extensions involved are regular For this purpose, the *frequency* of each morphological extension (*i e.*, number of occurrence in different decompositions) is computed. A threshold value for this count is chosen so that only those morphological extensions that have a frequency higher than the threshold are retained. The experimental results showing the effects of such a frequency threshold are summarised in Table 4.5 and shown graphically in figure 4.2.

Total number of distinct words 20140
Actual number of suffixes present 187

Morph' Extn' frequency threshold	B	C	Q	S	Precision (%)	Recall (%)	*f-measure'* (%)
1	10658	2218	654	185	1 71	98 93	3 35
2	866	494	353	173	16 65	92 51	28 22
3	281	214	247	155	35 55	82 89	49 76
4	154	123	186	142	47 97	75 94	58 80
5	84	74	144	131	60 93	70 05	65 17
6	57	45	119	121	67 98	64 71	66 30
7	40	36	101	113	73 86	60 43	66 47
8	30	29	86	108	78 26	57 75	66 46
9	26	24	78	101	79 53	54 01	64 33
10	22	14	71	93	80 87	49 73	61 59
11	18	13	68	88	83 02	47 06	60 07
12	14	10	64	80	85 11	42 78	56 94
13	10	6	59	74	88 10	39 57	54 61
14	8	4	55	69	89 61	36 90	52 27
15	7	3	52	65	90 28	34 76	50 19
16	5	3	46	63	92 65	33 69	49 41
17	2	2	46	60	96 77	32 09	48 19
18	2	2	44	59	96 72	31 55	47 58
19	2	1	43	56	96 55	29 95	45 71
20	1	0	41	54	98 18	28 88	44 63
21	0	0	40	53	100 00	28 34	44 17
22	0	0	38	50	100 00	26 74	42.19

S Suffix, Q Suffix-sequence, C Compound parts,
B Invalid morphological extension

Table 4 5· Effect of frequency of morphological extension in selecting valid suffixes

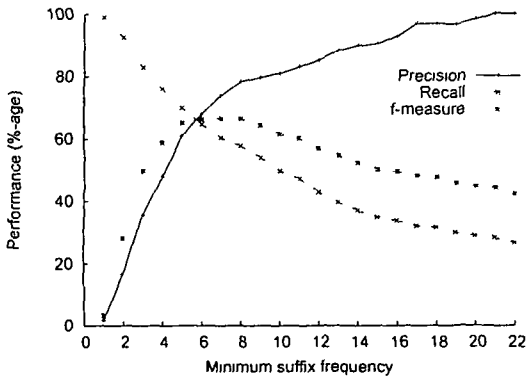


Figure 4.2: Effect of frequency in suffix selection

4.8.2 Base length

Many of the invalid decompositions in D involve *short* bases, *i.e.*, bases with very few letters. This is because a short word may match the leading portion of a longer word, even though the two may not be semantically related. Some examples are:

1.	$[der = de + r]_{\mathcal{A}}$	(দের)	/one-and-a-half/
2.	$[er = e + r]_{\mathcal{A}}$	(এর)	/leave (*imperative*)/
3.	$[mAt = mA + t]_{\mathcal{A}}$	(মাত)	/voice/
4.	$[bAHr = bA + Hr]_{\mathcal{A}}$	(বাহর)	/camp/
5.	$[deHr = de + Hr]_{\mathcal{A}}$	(দেহর)	/of body/
6.	$[kıHr = kı + Hr]_{\mathcal{A}}$	(কিহর)	/of what/.

$de_{\mathcal{A}}$ = give (*imperative*), $mA_{\mathcal{A}}$ = mother, $bA_{\mathcal{A}}$ = or, $kı_{\mathcal{A}}$ = what

In the above examples, all the decompositions are invalid. In 1 and 3, the bases as well as the morphological extensions are individually valid; in 2 the base is not valid; in 4, 5 and 6 the morphological extensions are not valid. Invalid bases comprising 1 or 2 letters are usually due to abbreviations, or letters used to enumerate points in the text. To avoid such invalid decompositions providing morphological extensions, which are often spurious, we try imposing a lower limit on the length of the bases of the decompositions that are considered for suffix acquisition. It may be mentioned here that computing the length of words or

portions of words must be carefully done since most of the prevalent encoding schemes for Assamese script use a non-uniform length of representation for the different letters. For example, in Unicode special characters are inserted to indicate the formation of ligatures, in a font-based encoding the single letter ৰা is realised by the sequence of the symbols ৰ and া, and in the Roman script based encoding we have used, the letter খ is represented by the string *kh*. Table 4.6 summarizes the effect of rejecting decompositions according to lengths of bases involved, and figure 4.3 presents it graphically.

Total number of distinct words 20140
Actual number of suffixes present· 187

Min Base Length	B	C	Q	S	Precision (%)	Recall (%)	*f-measure* (%)
1	10658	2218	654	185	1 71	98 93	3 35
2	5930	2105	643	182	2 98	97 33	5 78
3	2371	1726	561	177	6 95	94 65	12 94
4	834	1076	446	157	15 84	83 96	26 66
5	285	423	300	140	32 94	74 87	45 75
6	131	192	209	115	46 75	61 50	53 12
7	65	73	136	80	55 17	42 78	48 19
8	27	17	94	49	64 47	26 20	37 26
9	14	6	49	36	72 00	19 25	30 38
10	4	2	26	27	87 10	14 44	24 77
11	3	0	13	15	83 33	8 02	14 63
12	2	0	0	11	84 62	5 88	11 00
13	0	0	0	6	100 00	3 21	6 22

S Suffix, Q Suffix-sequence, C Compound parts,
B Invalid morphological extension

Table 4.6: Effect of base length (all letters) in selecting valid suffixes

Our intuition is that longer words are more *semantically stable* than shorter words. That is, if a word can be obtained by concatenating some letters to another word, the likelihood that they are semantically related is proportional to the length of the latter. We feel that this stems from the fact that morphology of a language and semantics of words are actually based on the spoken form of the language Longer words usually mean longer sequence of phonemes, and a long sequence of phonemes is more likely to be semantically unambiguous. Most scripts, however, do not reflect the actual phonetic length of words For example, the words "that" and "bat" has the same number of phonemes, though the first

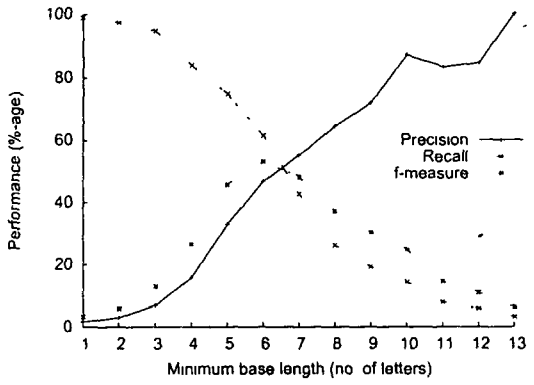


Figure 4.3: Effect of base length (all letters) in suffix selection

has four letters and the second has three. Similarly in Assamese, for instance, the words $path_{\text{অ}}$ (পথ /path/) and $kATh_{\text{অ}}$ (কাঠ /wood/) have the same number of phonemes, but the number of letters is different. In Assamese script, which is largely phonetic, this anomaly arises mainly because of the vowel $a_{\text{অ}}$. Unlike the other vowels in the script, $a_{\text{অ}}$ does not have a corresponding operator symbol. In some cases its implicit presence is assumed while in others it is not. In the word পথ this vowel is assumed to be present with the first letter প, but not with the second letter থ. In কাঠ the vowel operator া (corresponding to the vowel আ) is explicitly indicated, and no implicit vowel operator is assumed. In view of this, we use the following criteria to obtain a rough approximation of the *phoneme count*–

1. Each consonant is a phoneme. Each consonant in a ligature is counted independently.
2. Each vowel that occurs at the beginning of a word or after another vowel is a phoneme.

The effects of the selection of decompositions based on the phoneme length of the bases is summarized in Table 4.7, and shown graphically in figure 4.4.

Selecting decompositions based on the length of bases is one of the important criteria in the method proposed in [16] (see section 4.6.1). The value $p = 5$ as proposed there for a simple letter count (as against phoneme count) seems to be a fairly reasonable.

Total number of distinct words . 20140

Actual number of suffixes present: 187

Min. Base Length	B	C	Q	S	Precision (%)	Recall (%)	*f-measure* (%)
1	10658	2218	654	185	1.71	98.93	3.35
2	3255	1994	617	180	5.24	96.26	9.94
3	827	1095	420	155	15.78	82.89	26.52
4	255	350	257	127	33.25	67.91	44.64
5	93	106	159	87	48.33	46.52	47.41
6	27	17	87	48	64.00	25.67	36.64
7	10	5	45	36	78.26	19 25	30.90
8	3	2	12	15	83.33	8.02	14.63
9	0	0	5	11	100.00	5.88	11.11
10	0	0	0	6	100.00	3.21	6.22

S· Suffix; Q: Suffix-sequence; C: Compound parts,
B: Invalid morphological extension

Table 4.7: Effect of base length (phonemes) in selecting valid suffixes

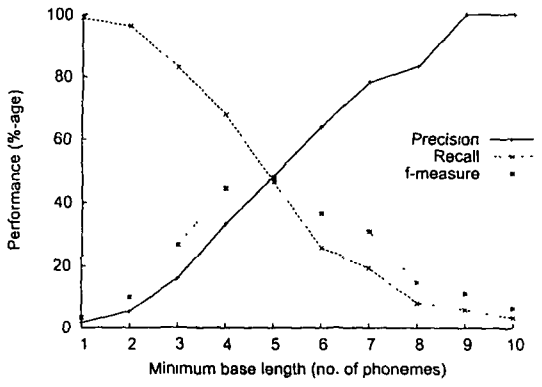


Figure 4.4: Effect of base length (phonemes) in suffix selection

wherever required, we use the term *segmented corpus* to refer to the corpus as comprising multiple articles, and the term *combined corpus* to refer to the single corpus obtained by merging the individual articles. The segmented corpus helps us avoid invalid decompositions such as

$$[kalaH = kal + aH]_A \qquad (\text{কলহ}) \qquad /\text{pot}/, \qquad \text{where } kal_A = \text{banana},$$

if the words $kalaH_A$ and kal_A do not occur in the same discourse. Decompositions identified in this way from several articles, are put together, and other selection criteria can be applied to these. For the corpus A of newspaper articles mentioned earlier, the results obtained are summarised in Table 4.8.

Number of newspaper articles		231
Number of input words	.	116096
Average number of input words per article		502
Actual number of suffixes present in the input	:	187
Number of distinct decompositions		8585
Number of distinct morphological extensions		2791
Distinct {S 154, Q 362, C 794, B.1481}		
Precision		18 84 %
Recall		82 35 %
f-measure		16 90 %

S Suffix, Q· Suffix-sequence, C Compound parts,
B Invalid morphological extension

Table 4.8: Summary of article-by-article decomposition of words

The results of the article-by-article decomposition exercise is along expected lines. There is a vast improvement of precision from 22.29% (considering the proportion of non-invalid morphological extensions to the total number of morphological extensions produced) to 46.94 %. The recall, however, shows a significant decline from 98.93% to 82.35%. Thirty-two suffixes that were detected in the initial decompositions, are missed in the article-by-article decompositions. This is because a morphological extension, x_S, goes undetected if no article contains two words w_1 and w_2 such that

$$[w_1 = w_2 + x].$$

In the combined corpus x_S is detected even if w_1 and w_2 occur in two different articles.

4.9 Combination of selection criteria

From the preceding discussion, it is seen that suitable combination of multiple selection criteria for morphological extensions is likely to give better performance than any single criterion. First we take a look at the effect of frequency (occurrence counts) of morphological extensions thresholds in article-by-article decompositions. The results are summarised in table 4.9 and shown graphically in figure 4.5.

Actual number of suffixes present: 187

Morph' Extn' frequency threshold	B	C	Q	S	Precision (%)	Recall (%)	*f-measure* (%)
1	1481	794	362	154	9.42	82.35	16.90
2	66	113	142	124	65.26	66.31	65.78
3	15	39	102	105	87.50	56.15	68.40
4	6	18	76	86	93.48	45.99	61.65
5	1	12	60	73	98.65	39.04	55.94

S: Suffix, Q: Suffix-sequence, C: Compound parts,
B: Invalid morphological extension

Table 4.9: Effect of frequency of morphological extension in selecting valid suffixes from article-by-article decompositions

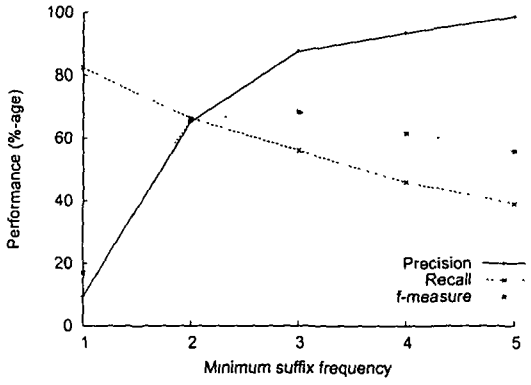


Figure 4.5: Effect of suffix frequency threshold over article-by-article decompositions

In table 4.9 we find that the occurrence count of the even valid morphological extensions is low in the article-by-article decomposition, and hence it is difficult to insist on a strong frequency of the morphological extension. But since the

prevalence of invalid morphological extensions is low in the article-by-article decompositions, we tried out other combinations of criteria over them. Tables 4.10 and 4.11 summarizes effects of base length thresholds. These results are shown graphically in figures 4.6 and 4.7 respectively.

Actual number of suffixes present: 187

Base length threshold	B	C	Q	S	Precision (%)	Recall (%)	*f-measure* (%)
1	1481	794	362	154	9.42	82.35	16.90
2	1201	790	359	154	11.37	82.35	19.97
3	397	660	302	149	27.29	79 68	40.65
4	155	470	229	128	45.23	68 45	54.47
5	59	200	164	101	63.12	54.01	58 21
6	35	107	125	82	70.09	43.85	53 95
7	20	52	81	56	73.68	29.95	42.59
8	5	7	50	40	88.89	21.39	34.48
9	2	3	27	29	93.55	15 51	26.61
10	0	1	15	19	100.00	10.16	18.45

S: Suffix; Q: Suffix-sequence; C: Compound parts;
B. Invalid morphological extension

Table 4.10: Effect of length (letter count) of base in selecting valid suffixes from article-by-article decompositions

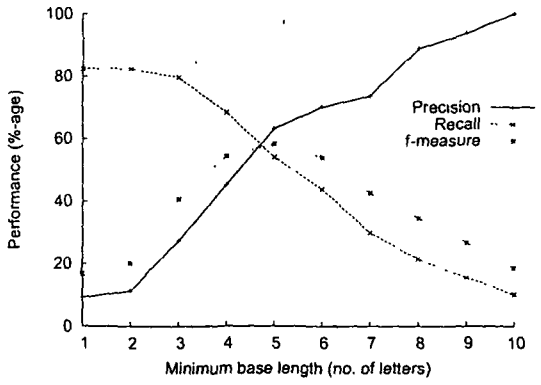


Figure 4.6: Effect of base length (all letters) threshold over article-by-article decompositions

Table 4.12 summarizes the effects of combinations of thresholds of base-length and frequencies of morphological extensions[4], in selection of suffixes from article-

[4]Number of occurrences in the decompositions from the combined corpus

Actual number of suffixes present. 187

Base length threshold	B	C	Q	S	Precision (%)	Recall (%)	*f-measure* (%)
1	1481	794	362	154	9.42	82.35	16.90
2	529	754	342	153	22 43	81.82	35.21
3	157	468	220	126	44.52	67.38	53.62
4	52	177	151	93	64.14	49 73	56.02
5	20	49	96	68	77.27	36.36	49.45
6	5	9	53	38	88 37	20.32	33.04
7	2	2	24	28	93.33	14.97	25.81

S· Suffix; Q: Suffix-sequence, C: Compound parts;
B Invalid morphological extension

Table 4.11: Effect of length (phoneme count) of base in selecting valid suffixes from article-by-article decompositions

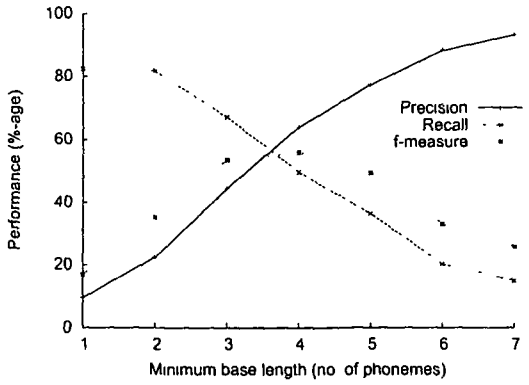


Figure 4.7: Effect of base length (phonemes) threshold over article-by-article decompositions

by-article decompositions. We see that simply using a threshold occurrence count of, say 3, gives a fairly good result even if the base length is ignored (*i.e.*, minimum base length is 1).

Number of articles . 231
Total number of distinct words : 20140

Base-length*	Suffix freq'** threshold	B	C	Q	S	Precision (%)	Recall (%)	*f-measure* (%)
1	1	1481	794	362	154	9 42	82.35	16.90
1	2	276	281	257	152	35.51	81.28	49.43
1	3	140	148	202	142	50.35	75.94	60.55
1	4	86	97	164	135	61.09	72.19	66.18
1	5	52	61	133	126	70.79	67.38	69.04
1	6	38	38	117	119	75.80	63.64	69.19
1	7	30	31	98	112	78.87	59.89	68.09
2	1	529	754	342	153	22.43	81.82	35.21
2	2	161	260	245	150	48.23	80.21	60.24
2	**3**	**92**	**141**	**197**	**140**	**60.34**	**74.87**	**66.83**
2	4	63	93	160	134	68.02	71.66	69.79
2	5	46	60	129	125	73.10	66.84	69.83
2	6	34	37	114	118	77.63	63.10	69.62
2	7	27	30	97	111	80.43	59.36	68.31
3	1	157	468	220	126	44.52	67.38	53.62
3	2	59	176	167	125	67.93	66.84	67.39
3	3	35	102	143	119	77.27	63.64	69.79
3	4	25	71	118	114	82.01	60 96	69 94
3	5	19	46	101	107	84.92	57.22	68.37
3	6	16	28	94	101	86.32	54.01	66.45
3	7	14	23	81	97	87 39	51.87	65.10

S· Suffix; Q: Suffix-sequence, C: Compound parts; B: Invalid morphological extension
* phoneme count; ** Occurrence in D, decompositions of combined corpus

Table 4.12: Combined selection criteria for morphological extension from article-by-article decompositions

One combination of criteria that is found to provide better results than other combinations considered is–

Minimum base length : 2 phonemes,
Minimum frequency of morphological extensions : 3 in decomposition of combined corpus.

The result of the above criteria for obtaining the suffixes from the same input corpus A considered so far, can be summarised as–

B = 92, C = 141, Q = 197, S = 140,
Recall = 74.87%, Precision = 60.34%, *f-measure* = 66.83%.

We consider this result better, because the *f-measure* is about the best, and recall value is good. In table 4.5 we find a slightly better *f-measure* for specific values of *morphological extension frequency threshold*, but the recall in those cases is below 65%. This lends credence to our intuition that

- two words that have identical leading portions are more likely to be semantically related if they occur in the same discourse, than if they do not;
- the phoneme count in words as defined by us in section 4.8.2, provides a good measure of semantic stability of words.

Thus, in general to select morphological extension the steps are

1. Obtain the initial decompositions D from the combined corpus.
2. Obtain the initial decompositions D_a article-by-article from the corpus.
3. From D_a retain the decompositions in which the bases have *two or more phonemes*, and, the morphological extensions have occurred f (*say,* three) or more times in D.

The threshold occurrence count f depends on the size of the corpus. Empirically it is seen that a good estimate is-

$$\begin{aligned} & f = 2, && n <= 50000, \\ \text{and,} \quad & f = \lceil \tfrac{n}{50000} \rceil, && n > 50000, \end{aligned}$$

where n is the number of words in the input corpus.

We observe that, over a set of criteria as above, other criteria do not significantly contribute to the selection of valid morphological extensions. Further, when morphological extensions with very high occurrences only, are retained (Table 4.5), the number of invalid morphological extensions falls to very low levels. This implies that when the selected morphological extensions are subsequently used for morphological analysis, *only a small number of words are decomposed using invalid morphological extensions.* A majority of the words

are properly decomposed. Hence, we simply need a threshold occurrence count for selection of morphological extensions that gives a *low (not necessarily, zero)* number of invalid morphological extensions.

4.10 Identifying compound parts

We have so far ignored the case of compound parts in the process of selection of valid morphological extensions. It is generally observed that in languages such as Assamese, two or more words that *frequently occur in a fixed order* are often merged to form compounds. Frequent fixed order occurrence of a pair of words usually implies that the words are directly related to each other in some way, in that expression, such as, one word qualifies the other, the two may be of identical function, *etc.* For example,

$[p\_rdhAn + m/nt\_rI]_A$ (প্রধানমন্ত্রী) /prime-minister/,
$[lrA + chowAlI]_A$ (লৰাছোৱালী) /boy (and) girl/.

Among the sequence of words that combine to form a compound, usually, only the last may be a derived word, *i.e.*, suffixes add to the compound as a whole.

From the results presented in sections 4.7, 4.8, and 4.9, we observe that a majority of suffixes have high occurrence counts and thus can be distinguished from the rest. By insisting on high value of the occurrence count of morphological extensions, the number of compound parts in the selected list of morphological extensions can be brought down. But doing so brings down the recall value of suffix identification. For the less frequently occurring suffixes some other detection criteria is required to distinguish from compound parts.

An intuitive criterion for distinguishing a compound part from other morphological extensions is that compound parts are likely to occur as independent words or in some other derived form, in the corpus if the corpus is sufficiently large. For example, the part m/nt_rI_A in the above example, or its other forms, such as m/nt_rIr_A, occur in the corpus. However, in case of Assamese, this criterion is not suitable in situations where suffixes are also, optionally, written detached from the base, thereby making a suffix appear to be a compound part. The following examples illustrate this

1.	$[m/nt\_rI + pd]_A$	(মন্ত্রীপদ)	/minister's post/
2.	$[m/nt\_rI + grAkI]_A$	(মন্ত্রীগৰাকী)	/the minister/
3.	$(m/nt\_rI\ \ grAkI)_A$	(মন্ত্রী গৰাকী)	/the minister/
4.	$(ghrr\ \ grAkI)_A$	(ঘৰৰ গৰাকী)	/house's owner/
5.	$[lrA + To]_A$	(লৰাটো)	/the boy/
6.	$(sklo\ \ chAt\_r)_A$	(সকলো ছাত্ৰ)	/all the students/
7.	$(chAt\_r\ \ sklo)_A$	(ছাত্ৰ সকলো)	/the students too/
8.	$[chAt\_r + sklo]_A$	(ছাত্ৰসকলো)	/the students too/.

In the above examples, pd_A is a compound part. The morpheme $grAkI_A$ in 2 and 3 plays the role of the determiner "the" like To_A in 5. To_A can only be a suffix, and hence $grAkI_A$ in 2 and 3 is a suffix. But $grAkI_A$ in 4 means "owner", and cannot be attached to the preceding word. $sklo_A$ in 7 and 8 have the same meaning and should be treated as a suffix. In fact, it is the suffix-sequence $(skl + o)_A$. But in 6, $sklo_A$ is an independent root word with a different meaning.

The above cases show that simple word-decomposition and morpheme-occurrence analysis may not provide a comprehensive mechanism to detect compound parts. On the other hand, we realise that for the ultimate goal of identifying the structure of words, we do not lose anything if we continue to treat compound parts as suffixes. Even if we do not distinguish between compound parts and suffixes or suffix-sequences, the decompositions of words are valid. Such decompositions of compounds can help in their recognition if the constituent parts are there in the lexicon. Hence, *we continue to consider compound parts and suffixes selected from the initial decompositions, alike for the purpose of morphological analysis of words, and for determining additional attributes of words we consider only the very frequently occurring suffixes.*

4.11 Suffix-sequences

A language such as Assamese allows certain suffixes to occur together in sequence in words. For example, suffixes s_1, s_2 and s_3 may occur with a base β as $\beta s_1 s_2 s_3$. We call a morphological extension comprising multiple suffixes a *composite suffix.*

The constituent suffixes of a composite suffix may or may not appear in other arrangements. For example,

1.	$[lrA + To + k]_{\mathcal{A}}$	(লৰাটোক)	/the boy (*accusative*)/
* 2.	$[lrA + k + To]_{\mathcal{A}}$		
3.	$[kitAp + r + khini]_{\mathcal{A}}$	(কিতাপৰখিনি)	/(the contents) of book/
4.	$[kitAp + khini + r]_{\mathcal{A}}$	(কিতাপখিনিৰ)	/the books'/.

In the second example marked "*", the suffix-sequence $[k + To]_{\mathcal{A}}$ is not valid. Examples 3 and 4 have the same suffixes in alternate arrangements. The implications of the suffixes in the different arrangements are different.

Unless composite suffixes are decomposed into the sequence of suffixes, they would appear to be single suffixes and make the set of suffixes unduly large. Identifying the individual suffixes in a composite suffix provides the same kind of benefits as obtained upon breaking up a word into a base and morphological extension. With the knowledge of a small set of suffixes, a much larger set of composite suffixes can be recognized. It provides a *structured* way to discover the attributes of words. When a word contains a sequence of suffixes, the morphological analysis of the word is complete only if all the parts of the sequence are identified.

In chapter 6, we discuss approaches for classification of words based on use of suffixes. In such efforts, recognizing all the constituent suffixes in words instead of composite suffixes, is very useful. Here is a simple illustration:

1.	$[kukur + Tok]_{\mathcal{A}}$	(কুকুৰটোক)	/the dog (*accusative*)/
2.	$[crAi + To]_{\mathcal{A}}$	(চৰাইটো)	/the bird/.

Here, unless the composite suffix $Tok_{\mathcal{A}}$ (in 2) is recognized as $(To + k)_{\mathcal{A}}$ we may not realize that the words $kukur_{\mathcal{A}}$ and $crAi_{\mathcal{A}}$ are of the same category.

We denote a suffix-sequence comprising the non-$NULL$ suffixes $s_1, s_2, ...s_3$ in that order as $s_1 + s_2 + ... + s_n$. A suffix-sequence comprising only the $NULL$ suffix is referred to as the *NULL suffix-sequence.* We call the decomposition of a word a *complete decomposition* if the decomposition is valid and none of the parts in the decomposition can be further decomposed into multiple parts. If a part in a decomposition can be further decomposed, we call it a *composite part.* If the base

comprises a single morpheme, it is a *root*. We call the number of non-NULL parts in a decomposition beyond the first part, the *degree of the decomposition*. Thus a *trivial decomposition* (see section 4.5) has degree 0, and a complete decomposition of a word has the highest possible degree of decomposition for that word.

4.11.1 Identifying suffix-sequences

Suffix-sequences can be identified by successively replacing the base of a decomposition by a possible *non-trivial* decomposition of it, as long as such a replacement is possible. That is, if $[w_i = \beta_i + p_i]$ and $[w_j = \beta_j + p_j]$ are two decompositions and $(\beta_i = w_j)$, a combined decomposition can be written as

$$[w_i = \beta_j + p_j + p_i] \ ,$$

where we get $(p_j + p_i)$ as a suffix-sequence. We refer to this process as *recursive reduction of the bases* Suppose, we get the following decompositions by these steps:

$$\begin{aligned} & [\omega_i = \beta_i + p_1 + p_2 + p_3], \\ \text{and} \quad & [\omega_j = \beta_j + p_1 + p_2]. \end{aligned}$$

These two decompositions contain the two suffix-sequences $(p_1 + p_2 + p_3)$ and $(p_1 + p_2)$, where the latter is actually a subsequence of the former. Since subsequences of a suffix-sequence are also valid suffix-sequences, we may record only those suffix-sequences that are not sub-sequences of any other suffix-sequence, as long as the subsequences are valid.

In practice, there can be multiple distinct decompositions for same words, using different base-suffix pairs. This can make recursive reduction of bases problematic. Hence, before recursive reduction of bases is performed, the multiple distinct decompositions of same words are *unified* as described in section 4.11.4. A simple implementation of the steps to identify the suffix-sequences from a given set of initial decompositions is given in section C.3. We perform this exercise over the initial set of decompositions obtained as described in section 4.7, involving morphological extensions obtained using the criteria described in section 4.9. The suffix-sequences obtained can be qualitatively classified as

A. correctly identified, *eg.*, the suffix-sequence $(A + b + lE)_{\mathcal{A}}$ in the decomposition

$[krAblE = kr + A + b + lE]_{\mathcal{A}}$ (কৰাবলৈ) /to get done/,

B. correctly identified, but needs further decomposition, *eg.*, the suffix-sequence $(A + znk)_{\mathcal{A}}$ in the following decomposition should actually have been $(A + zn + k)_{\mathcal{A}}$

$[krAznk = kr + A + znk]_{\mathcal{A}}$ (কৰাজনক) /one who does/,

C. correct but identified in inappropriate decompositions only. *eg.*, the suffix-sequence $(A + zn + r)_{\mathcal{A}}$ is valid but the following decomposition from which it has been obtained is not valid

$[mHAznr = mH + A + zn + r]_{\mathcal{A}}$ (মহাজনৰ) /shop-owner's/,

D. correct but needs further decomposition and identified in inappropriate decompositions only. *eg.*, the suffix-sequence $(A + zne)_{\mathcal{A}}$ should actually be $(A + zn + e)_{\mathcal{A}}$, and it has been obtained from the following decomposition, which is not valid

$[mHAzne = mH + A + zne]_{\mathcal{A}}$ (মহাজনে) /shop-owner (*ergative*)/,

E. incorrect, *eg.*, the suffix-sequence $(Ai + bor)_{\mathcal{A}}$ in the following decomposition is not valid

$[ThAibor = Th + Ai + bor]_{\mathcal{A}}$ (ঠাইবোৰ) /the places/.

A suffix-sequence may be incorrect either because one or more of its constituents is not a valid suffix part, or the break-up of the sequence is not correct. If a suffix-sequence is incorrect, then all suffix-sequences of which it is a subsequence, are also incorrect, but, some of its subsequences may be correct. Hence, *for the purpose of a quantitative analysis* where we count the number of valid and invalid suffix-sequences, a suffix-sequence x_i, should not be dropped due to the presence of a longer sequence x_j, even if x_i is a subsequence of x_j. Further, in such an analysis, we count the correct sequences ignoring the fact that some of them may be obtained from inappropriate decompositions only. That is, we count type A and type C decompositions together, and type B and type D decompositions together.

The outcome of our suffix-sequence identification exercise is summarised in Table 4.13. The column headings A, B, C, D, and E refer to the qualitative classification mentioned earlier in this section. The column "A+C" gives the count of suffix-sequences that are correct and completely decomposed, the column "B+D" gives the count of suffix-sequences that are correct but require further decomposed, and the column "E" gives the count of incorrect suffix-sequences. The first row gives the numbers when the only restriction applied is that the bases of the decompositions must have at least one phoneme. Since there is much room for improvement, the subsequent rows give the outcome with different values of the *minimum base length* and the *minimum frequency* (number of occurrences) of the suffix-sequences.

Total number of distinct words : 20140

Min base length	Min suff.-seq. frequency	No. of suff.-seq identified		
		A+C	B+D	E
1	1	638	470	2468
1	2	352	240	387
1	3	260	173	156
1	4	212	140	100
1	5	172	114	70
2	1	555	411	636
2	2	325	214	170
2	3	258	161	76
2	4	207	131	43
3	1	399	293	235
3	2	239	161	44
3	3	185	123	17
4	1	276	207	129
4	2	158	108	16

Table 4.13: Initial identification of suffix-sequences

While increasing the restrictions reduces the number of incorrect suffix-sequences identified (column E), the number of valid suffix-sequences also gets reduced. One important observation in the results is that most of the incorrect suffix-sequences actually has some common defective subsequences. For example, the suffix-sequences– $(A+kt+khn+e)_A, (A+kt+khn+k)_A, (A+kt+khn+r)_A, (A+kt+khn+t)_A, (A+kt+khne)_A, (A+kt+khnk)_A, (A+kt+khnr)_A, (A+kt+khnt)_A, (A+kt+lE)_A, (A+kt+r+e)_A, (A+kt+re)_A, (A+kt+t+e)_A$

and $(A+kt+te)_A$ all are incorrect due to the presence of the invalid subsequence $(A+kt)_A$. On the other hand, many of the suffix-sequences that need further decompositions (column "B+D"), are adequately decomposed in some other word decompositions. For example, while there are the type B suffix-sequences $(I+skle)_A, (I+sklk)_A$ and $(I+sklr)_A$, there also are the type A suffix-sequences $(I+skl+e)_A, (I+skl+k)_A$ and $(I+skl+r)_A$. The reason there are sequences that need further decomposition, is simply that the suffix list used to obtain the suffix-sequences contains elements such as $sklk_A, skle_A, etc.$, which are actually suffix-sequences themselves. These elements crept into in the suffix list because our unsupervised method used to prepare the list fails to prevent some such instances. In the following sections, we discuss some approaches to make the suffix-sequence identification more effective.

4.11.2 Alternative suffix-sequences

Two suffix-sequences, $(x_a{=}x_{a_1}{+}x_{a_2}{+}...{+}x_{a_m})$ and $(x_b{=}x_{b_1}{+}x_{b_2}{+}...{+}x_{b_n})$ are *alternative suffix-sequences* with respect to each other if upon concatenation they produce identical strings. That is,

$$(x_{a_1}x_{a_2}...x_{a_m} = x_{b_1}x_{b_2}...x_{b_n}) \ .$$

We denote this relationship between x_a and x_b as

$$(x_a =_a x_b) \ .$$

For example,

$$(I+sklk)_A \ =_a \ (I+skl+k)_A \ .$$

4.11.3 Alternative decompositions

The suffix-sequence identification process described above does not necessarily produce complete decomposition of the words. Due to the nature of the process for obtaining the initial decompositions, there may be several different decompositions for the same word. If δ_1 and δ_2 are distinct decompositions of the word ω, we term them *alternative decompositions*, and represent this relationship too, as

$$\delta_1 =_a \delta_2 \ .$$

where,

$$\begin{aligned} \delta_1 : \quad & [\omega = \beta_1 + x_1] \\ \delta_2 : \quad & [\omega = \beta_2 + x_2] \ . \end{aligned}$$

If two alternative decompositions of a word involve the same base, then the two suffix-sequences are alternative suffix-sequences That is, in the two decompositions above, if ($\beta_1 = \beta_2$), then,

$$(x_1 =_a x_2) \ .$$

On the other hand, if the bases involved in two alternative decompositions of a word are distinct, we call the decomposition with the longer base *shallower* than the other, and the suffix-sequence in the former is *shallower* than that in the latter. That is, if,

$$|\beta_1| \leq |\beta_2| \ ,$$

then, δ_2 is shallower than δ_1, and x_2 is shallower than x_1. We denote this *shallower* relationship as

$$\begin{aligned} & \delta_2 <=_s \delta_1 \ , \\ \text{and} \quad & x_2 <=_s x_1. \end{aligned}$$

In the suffix-sequence identification process, we can obtain n alternative decompositions from a given decomposition δ, if there are n alternative decompositions for the base β, of δ. That is, if

$$\begin{aligned} \delta : \quad & [\omega = \beta + x] \ , \\ \delta_1 : \quad & [\beta = \beta_1 + x_1] \ , \\ \delta_2 : \quad & [\beta = \beta_2 + x_2] \ , \\ & \vdots \\ \delta_n : \quad & [\beta = \beta_n + x_n] \ , \end{aligned}$$

then,

$$\begin{aligned} & [\omega = \beta_1 + x_1 + x] \ , \\ & [\omega = \beta_2 + x_2 + x] \ , \\ & \vdots \\ & [\omega = \beta_n + x_n + x] \ , \end{aligned}$$

are n alternative decompositions of ω.

4.11.4 Unification of decompositions

When there are multiple alternative decompositions for a word, they can be combined to obtain a single decomposition. For this we generate a decomposition with *partition points* (see section 4.5) at all points in the original string of the word, where any of the alternative decompositions has a partition point. We call this process *unification of the decompositions*. For instance, suppose the first alternative decomposition of ω has partition points at offsets 3 and 7, and the second has partition points at offsets 5 and 7. Upon unification, we have a decomposition with partition points at 3, 5, and 7 with respect to ω. The resultant unified decomposition has a degree at least as high as the highest degree among the alternative decompositions. A simple implementation of the process of unifying decompositions is described in section C.2.

As an example of unification of decompositions, suppose we have the initial decompositions each of degree 1:

$$[sbhAkhnr = sbhA + khnr]_{\mathcal{A}} \quad (\text{সভা} + \text{খনর}) \quad /\text{of the meeting}/,$$
$$[sbhAkhnr = sbhAkhn + r]_{\mathcal{A}} \quad (\text{সভাখন} + \text{র}) \quad /\text{of the meeting}/.$$

Then the unified decomposition is:

$$[sbhAkhnr = sbhA + khn + r]_{\mathcal{A}} \quad (\text{সভা} + \text{খন} + \text{র}) \quad /\text{of the meeting}/.$$

which is of degree 2. The unified decomposition contains the part $khn_{\mathcal{A}}$ which was not there in the given alternative decompositions.

The unification of decompositions does not necessarily produce a *complete decomposition* of the word. However, it is generally a safe way to obtain a higher degree decomposition of words, and possibly, discover new parts. It is safe because, for the given word, no new partition points are introduced. So if the given alternative decompositions are valid, the unified decomposition has valid partition points.

A decomposition implies the existence of the different words which may be obtained by adding to the base zero or more parts of the morphological extension. That is, the decomposition

$$\delta : [\beta + x_1 + ... + x_n]$$

implies the existence of the decompositions, δ_i such that

$$\delta_i : \quad [\beta + x_1 + ... + x_i], \quad 1 \leq i \leq n,$$
$$[\beta + NULL], \quad i = 0 .$$

We represent this relationship as

$$\delta_i <= \delta,$$

which means that δ_i and the word that it represents can be *extracted* from δ. Alternatively, δ *generates* δ_i and the word that δ_i represents. For example,

$$\begin{aligned} [sbhA = sbhA + NULL]_{\text{অ}} &<= [sbhAkhnr = sbhA + khn + r]_{\text{অ}}, \\ [sbhAkhn = sbhA + khn]_{\text{অ}} &<= [sbhAkhnr = sbhA + khn + r]_{\text{অ}}, \\ [sbhAkhnr = sbhA + khn + r]_{\text{অ}} &<= [sbhAkhnr = sbhA + khn + r]_{\text{অ}}, \end{aligned}$$

where, $sbhA_{\text{অ}}$ (সভা) means *meeting*, $sbhAkhn_{\text{অ}}$ (সভাখন) means *the meeting*, and $sbhAkhnr_{\text{অ}}$ (সভাখনৰ) means *of the meeting*. The three words too are generated by the decomposition of $sbhAkhnr_{\text{অ}}$.

For compact representation of a set of decompositions, we may leave out a decomposition if it can be *extracted* from another distinct decomposition in that set. For example, if we have the decomposition for $sbhAkhnr_{\text{অ}}$ as shown above, then we may leave out the decompositions for $sbhAkhn_{\text{অ}}$, and $sbhA_{\text{অ}}$ (the trivial decomposition). We refer to this process of filtering out from a set decompositions that can be extracted from other decompositions in that set, as *compaction*.

4.12 Boundary adjustment in word decompositions

The suffix and suffix-sequence identification method discussed above is susceptible to certain tricky morphological phenomena. For instance, suppose there are the two suffixes, b_S, and bc_S in the language. Further, bc_S is not a composite suffix, *i.e.*, we cannot break it up as the sequence $(b + c)_S$. Now, if the corpus contains the words a_S, ab_S, and abc_S, we get the decompositions $[ab = a + b]_S$ and $[abc = a + b + c]_S$. That is, $(b + c)_S$ would be wrongly learnt as a suffix-sequence. The most prominent example of this phenomenon in Assamese is the case of the suffixes $r_{\text{অ}}$ and $rUpe_{\text{অ}}$ (ৰ, ৰূপে). These two suffixes frequently occur with the same roots, which are nouns. For example,

$[mAnuHr = mAnuH + r]_{\mathcal{A}}$	([মানুহৰ = মানুহ + ৰ])	/of human/
$[mAnuHrUpe = mAnuH + rUpe]_{\mathcal{A}}$	([মানুহৰূপে = মানুহ + ৰূপে])	/as a human/.

Here, the letter string $Upe_{\mathcal{A}}$, which is not a suffix is identified as one. This happens if the derivatives $mAnuHr$ and $mAnuHrUpe$ occur in the corpus.

To avoid spurious breaking up of the suffix bc_s into the sequence $(b + c)_s$, we need to note that if $(b + c)_s$ is really a suffix-sequence, c_s should have some occurrence independent of b preceding it. If every occurrence of c_s is preceded by b_s, there is no advantage of considering c_s individually instead of considering it as bc_s. Again, suppose there is some word, say gbc_s, which is decomposed as $[gbc = gb + c]_s$, and not as $[gbc = g + b + c]_s$ because the word g_s is not present in the corpus. In this case too, it is necessary to register the occurrence of b_s preceding c_s. In general, after the suffixes and suffix-sequences are identified according to the method described so far, for each suffix we check if all occurrences of the suffix have a common letter sequence preceding it. If so, the suffix should be extended to include that common letter sequence preceding it. We refer to this exercise as *suffix extension*.

4.13 Very irregular morphological extension parts

There are certain morphological extension parts, which are valid but hold only in very few cases, *i.e.*, they are not *regular*. For example, the decomposition

[clothe = cloth + e]

is valid but the morphological extension part *"e"* is not regular, in the sense that only in very few cases it adds to a base to give a valid derivative. Decompositions such as, [pathe = path + e] (*"e"* added to a valid base) or [caste = cast + e] (a valid word decomposed using *"e"* as morphological extension) are not valid. In Assamese, consider the following decompositions:

1. $[thiyE = thiy + E]_A$ ([থিয়ৈ = থিয় + ঐ]) /standing/
2. $[krilE = kril + E]_A$ ([কৰিলৈ = কৰিল + ঐ]) /after doing/
3. $[prilE = pril + E]_A$ ([পৰিলৈ = পৰিল + ঐ]) /after falling/
4. $[prılE = pri + lE]_A$ ([পৰিলৈ = পৰি + লৈ]) /after falling/
5. $[DAnGrkE = DAnGrk + E]_A$ ([ডাঙৰকৈ = ডাঙৰক + ঐ]) /loudly/
6. $[DAnGrkE = DAnGr + kE]_A$ ([ডাঙৰকৈ = ডাঙৰ + কৈ]) /loudly/
7. $[kE = k + E]_A$ ([কৈ = ক + ঐ]) /saying (*participle*)/
8. $[lE = l + E]_A$ ([লৈ = ল + ঐ]) /taking (*participle*)/.

Decompositions 1, 7 and 8 are valid, but the part E_A (ঐ) is not a regular suffix, *i.e.*, it is the valid suffix only in very few of the words where it occurs as the trailing part. Decompositions such as 2 and 3 involving this morphological extension, are not valid, though the derivatives are valid words. In 2, the base too, is invalid. A very tricky case in Assamese is the decomposition 3, where the derivative and the base are both valid words and are closely related semantically. But the decomposition is not valid as the derivative $prilE_A$ is not derived from the base $pril_A$. The correct decomposition is 4. Similarly, for the word $DAnGrkE_A$ the decomposition 6 is valid and 5 is not.

Due to the difficulty in dealing with such highly irregular morphological extension parts, we attempt to merge them with the preceding letters in the decompositions. This requires a heuristic more complex than the one mentioned for *suffix extension*, since the letters preceding the irregular morphological extension part in different decompositions are not identical. Hence, we use the criteria that an irregular morphological extension part has a comparatively low occurrence count (say, less than 3 times the required threshold count to accept a part) and merging it with one or more preceding letters of the decompositions produces some known morphological extension part that has a higher occurrence count[5]. In the example above, wherever E_A is preceded by l_A or k_A, merging them produces lE_A and kE_A respectively, which have higher occurrence counts than E_A. We refer to this step of merging as *suffix consolidation*.

[5]The occurrence count considered here is that before unification of decompositions.

4.14 Orthographic peculiarities

Concatenative morphology is a phenomenon that originates from fusing adjacent morphemes according to the convenience of *speakers* of the language. At the point of fusion the pronunciation is sometimes represented in the written form by a changed spelling instead of the concatenation of the basic spelling of the fused morphemes. For example, the actual suffix in the following decomposition should be $e_{\mathcal{A}}$

$$[gruwe = gru + we]_{\mathcal{A}} \quad [\text{গরুরে} = \text{গরু+ রে}] \quad /\text{cow } (ergative)/ \ .$$

This kind of spelling modification affects the identification of suffixes. In this example, the unsupervised suffix acquisition identifies $we_{\mathcal{A}}$ as a suffix, whereas the actual suffix involved is nothing but the more regular $e_{\mathcal{A}}$. In our method, we also fail to identify the presence of a suffix. For example, the actual base in the following decomposition is $kkAi_{\mathcal{A}}$

$$[kkAye = kkAi + e]_{\mathcal{A}} \quad [\text{ককায়ে} = \text{ককাই+ এ}] \quad /\text{brother } (ergative)/ \ .$$

The above examples represent peculiarities due to a script and the way it is used in a particular language. These pose difficulties in unsupervised acquisition of suffixes. We have not taken any step to deal with these difficulties. Some amount of supervision in the form of hand crafted rules to deal with such irregularities can make the process of morphology acquisition as well as morphological analysis more effective.

4.15 Consolidating the morphological features and building a lexicon

The discussion in the preceding sections, starting from 4.9 forms the basis for steps to consolidate the morphological knowledge. This knowledge comprises knowledge of suffixes, knowledge of suffix sequences, and knowledge of compounds. Knowledge of suffixes can be near-exhaustive, but possible suffix sequences and compounds can be many and it is unrealistic to expect to list them all through a computational process like the one we have discussed. To recognize suffix sequences and compounds beyond what is acquired from the

training corpus, we follow some generic criteria, which we discuss in chapter 5.

The morphological knowledge that we acquire is to be subsequently used for analysis of words of new texts, that we refer to as test input. Test input is most likely to be in not-so-large chunks, such as paragraphs, essays, articles, *etc.*, consisting of about few thousands of words. In such texts, several root words may occur only a very few number of times each, in the root form or some of the possible secondary forms. For example, suppose a sports news article contains the sentence-

$(bhArte\ Tenict\ eTA\ meDel\ lAbh\ krile)_{\mathcal{A}}$

ভারতে টেনিচত এটা মেডেল লাভ করিলে

/India won one medal in tennis/,

where the word $Tenic_{\mathcal{A}}$ occurs in an inflected form with a case marker $t_{\mathcal{A}}$, and it is the only occurrence of the word. To analyse the word $Tenict_{\mathcal{A}}$ some other occurrence of the root in some form is desirable. It would basically provide a *confidence* regarding the root. To meet this requirement a lexicon is required. A lexicon serves as a repository of knowledge of root words along with some attributes. We build a lexicon using the evidence of words seen in the training corpus. More specifically, we record the decompositions that we obtain for the words in the training corpus in the lexicon. The *initial decomposition* exercise does not produce all the possible decompositions for the words in the input corpus. To build the lexicon, we identify more decompositions of words using morphological extensions already acquired. This produces base words that were not there in the input. With these new words more decompositions can be identified, and this can be carried out as a boot-strapping process.

The use of a lexicon (or dictionary) invariably requires searching. The most common search key is a word, but search for specific sub-word morphemes is also required sometimes. Again, in many applications addition of new entries to a dictionary, and deletion of existing entries is continuously done. To facilitate dynamic addition and deletion of entries along with efficient search for existing entries suitable data structures, such as, AVL trees, hash tables, *etc.* can be used ([11]). AVL trees are binary search trees and searching an entry requires $O(\log_2 n)$ time, where n is the number of entries in the lexicon. Hash tables can give better performance, but both performance and space requirements can be unpredictable.

In our case, we assume that the lexicon built with the evidence from the "large" training corpus is static, and we do not update it dynamically. Hence, we prepare only for searching of entries, and maintain the lexicon as a linear sequence of entries sorted on the search key, that is, the word. A search in such a lexicon can be accomplished in $O(\log_2 n)$ time. Moreover, such a lexicon can be more easily rearranged[6] to facilitate efficient search on other keys. Conceptually, an entry of our lexicon has the following format-

<word> <base> <morphological extension> <other attributes>,

where "other attributes" is additional information such as category of the base, *etc.*, that may be added separately.

The analysis of a word as recorded in the lexicon depends on the presence of other related words (words derived from the same root, or words having similar morphological extensions) in the training corpus. If for a particular word, adequate number of related words are not present, its analysis may be incomplete or even incorrect. For example, if the lexicon contains the words *bipd*$_A$ (বিপদ /danger/) and *bipdznk*$_A$ (বিপদজনক /dangerous/), both together may be recorded as a single entry in the lexicon–

$$<bipdznk \quad bipd \quad zn+k>_A$$

which is incorrect. The correct morphological extension is *znk*$_A$. If the corpus contained the word *bipdznkbhAwe*$_A$ (বিপদজনকভাবে /dangerously/) too, the lexicon entry would have been

$$<bipdznkbhAwe \quad bipd \quad znk+bhAwe>_A$$

which is correct.

We summarize the steps that we follow, starting from the initial decomposition, to consolidation of the morphological knowledge acquired and creation of a lexicon, below. Let the set of words in the input corpus be W.

[6]sorted on another key

Stage 1. Prepare initial set of suffixes, S_1

First we obtain the initial set of suffixes according to the steps outlined in section 4.9.

1. Obtain the initial decompositions, D_{i_a}, for the words in the corpus article-by-article.
2. Obtain the initial decompositions, D_{i_c}, for the words in the combined corpus.
3. Perform suffix extension over the decompositions in D_{i_a} (see section 4.12).
4. Let S_1 be the set of morphological extensions that have occurred with bases with at least p (=2) phonemes in D_{i_a}, and have occurred in at least f distinct decompositions in D_{i_c}.

Stage 2. Get comprehensive set of decompositions, D

Next, we use the set of suffixes S_1 to further decompose the input words in a boot-strapping way.

1. Let $W_1 = W$ *i.e.*, the set of input words.
2. Obtain the set, D_1, of all possible decompositions $[w = b + s]_S$, such that $w \in W_1$ and $s \in (S_1 \cup \{NULL\})$.
3. Obtain a set of decompositions, D by selecting from D_1 those decompositions, which are either trivial or the base involved occurs in at least two decompositions, *i.e.*,

   if $[w_i = b + s_i] \in D_1$ then
   $D := D \cup \{[w_i = b + s_i]\}$ iff
   $s_i = NULL$, or, $\exists [w_j = b + s_j] \in D_1$, where $w_i \neq w_j$.
4. If there are bases involved in decompositions in D, which are *figured words*, *i.e.*, they are not in W_1, then include such bases in W_1 and *goto* step 2.
5. Perform suffix consolidation over the decompositions in D (see section 4.13).

Stage 3. Obtain higher degree decompositions, D_2

The initial set of suffixes S_1 contains several composite suffixes too. Hence some decompositions in D may have scope for further decompositions. We process them to decompose further to obtain a set of higher degree decompositions, D_2.

1. Initialize set D_2 by unifying decompositions in D (see 4.11.4). Due to unification some suffix-parts that are not there in S_1 may be produced.

2. Recursively reduce the bases of the decompositions in D_2 (see section 4.11.1). That is,

   if $\{[w = b_i + x_i], [b_i = b_j + x_j]\} \subset D_2$, and $x_j \neq NULL$, then
   $D_2 := (D_2 - \{[w = b_j + x_j]\}) \cup \{[w = b_j + x_j + x_i]\}$.

3. Perform *compaction* of the decompositions set D_2 (see section 4.11.4). That is,

   if $\{[w = b + x_i + x], [w_i = b + x_i]\} \subset D_2$, and $x_i, x \neq NULL$, then
   $D_2 := (D_2 - \{[w_i = b + x_i]\})$.

Stage 4. Verify new suffix-parts

Since S_1 is obtained from the initial decomposition exercise, each morphological extension in S_1 occurs as the final part of some input word. Suppose S_2 is the set of suffix-parts occurring in D_2. S_2 may contain *new* suffix-parts that are not in S_1. (A new suffix-part would always occur as a non-final part in an unified decomposition.) Since D_2 is originally taken from D, some of the decompositions in D_2 may be of *figured words*, *i.e.*, words not in W. A new suffix-part might be the final part of figured word. For example, suppose D_2 contains the decomposition

$$[sbhAkhnr = sbhA + khn + r]_{\text{অ}} \quad (\text{সভা} + \text{খন} + \text{র}) \quad \text{/of the meeting/}$$

due to unification of the following decompositions in D:

$$[sbhAkhnr = sbhA + khnr]_{\text{অ}} \quad (\text{সভা} + \text{খনর}) \quad \text{/of the meeting/},$$
$$[sbhAkhnr = sbhAkhn + r]_{\text{অ}} \quad (\text{সভাখন} + \text{র}) \quad \text{/of the meeting/}.$$

If the word $sbhAkhn_{\text{অ}}$ is a figured word, then the new suffix $khn_{\text{অ}}$ is the final part of the figured word. If a new suffix-part in S_2 occurs as the final part of

figured words only, then we eliminate that new suffix-part by merging it with the part following it in the decompositions where it occurs. For example, suppose D_2 contains the decompositions

1. $[crAiTi = crAiT + i]_A$ (চৰাইটি + ই) /the bird/,
2. $[crAiTo = crAiT + o]_A$ (চৰাইটি + ও) /the bird/,
3. $[crAiTo = crAi + To]_A$ (চৰাই+ টো) /the bird/.

where, $crAiT$ is a figured word and is, in fact, invalid. Then by unification of decompositions 2 and 3, we obtain

$$[crAiTo = crAi + T + o]_A.$$

Now, the new suffix part T_A is actually invalid, and would not occur as the final suffix-part of the decomposition of any real word. Since we do not find any input word whose decomposition has T_A as the final part, we merge T_A with the suffix part following it in decompositions 1 and 2, and obtain

$$[crAiTo = crAi + To]_A$$
$$[crAiTi = crAi + Ti]_A.$$

We state the above, more specifically as:

Suppose, $s \in (S_2 - S_1)$, *i.e.*, s is a new suffix part,

$\delta_j : [w_j = b_i + x_i + s + p_i + x_j]$ and $\delta_j \in D_2$, where x_i and x_j are, possibly NULL, parts-sequences, *i.e.*, δ_j is a decomposition involving s.

If $b_i x_i s \notin W \; \forall \delta_j$, (*i.e.*, all words with s as the final suffix part extracted from decompositions in D_2 are figured words) then for each δ_j, do

$$D_2 := (D_2 - \{\delta_j\}) \cup \{[w_j = b_i + x_i + sp_i + x_j]\}.$$

That is, merge s with the part following it in the decompositions.

Stage 5. Generate *more likely* alternative decompositions, D_3:

In building the lexicon, our primary lookout is to have decompositions that *are valid* and have a *high degree*[7]. For *validity* of a morphological extension, we define a threshold value, q, for the minimum occurrence count[8] of valid

[7]High degree implies more number of morphemes present are identified

[8]Occurrence is counted for distinct words formed.

morphological extensions. Empirically, we find that the suitable value of q is 3 for a corpus larger than 100,000 words. For each decomposition in D_2 we consider the alternative morphological extensions that contain all the partition points of the original morphological extension. For example, for the decomposition

$$[mAnuHznrprAHe = mAnuH + znr + prAHe]_{\mathcal{A}}$$
(মানুহজনৰপৰাহে) /only from the person/,

we get the additional decompositions

$$[mAnuHznrprAHe = mAnuH + zn + r + prAHe]_{\mathcal{A}}$$
$$[mAnuHznrprAHe = mAnuH + znr + prA + He]_{\mathcal{A}}$$
$$[mAnuHznrprAHe = mAnuH + zn + r + prA + He]_{\mathcal{A}}.$$

If such a morphological extension is not valid, we successively merge its initial parts with the base until the remaining morphological extension is valid or is $NULL$. For example, from the invalid decomposition

$$[bzArkhnrprAHe = bzA + r + khn + r + prA + He]_{\mathcal{A}}$$
(বজাৰখনৰপৰাহে) /only from the market/,

we get the valid decomposition

$$[bzArkhnrprAHe = bzAr + khn + r + prA + He]_{\mathcal{A}}.$$

From the alternative decompositions thus obtained, we select the one with the shortest base, and highest degree, *in that order*. If there are more than one such decompositions, we unify them.

We state the above more specifically as,

1. Suppose, $C(X)$ denotes the occurrence count of the morphological extension X, in D_2.
   Suppose, $\delta : [\omega = \beta + x]_S$ is a decomposition in D_2.

2 Find the decompositions

$$\delta_i : [\omega = \beta + x_i]$$

such that $x_i =_a x_S$ (*i.e.*, x_i is an alternative suffix-sequence of x_S).

3. If $C(x_j) < q$ (*i.e.* occurrence count of x_j is lower than threshold q):

   Suppose $x_j = (a_1 + a_2 + ... + a_n)$;
   then modify δ_j as

$$\delta_j : [\omega = \beta_k + x_{j_k}]$$

where, $x_{j_k} = (a_k + ... + a_n)$, $\beta_k = (\beta a_1 ... a_{k-1})$ and k is the smallest number such that

$$C(x_{j_k}) \geq q, \quad \text{and} \quad C(a_{k-1} + ... + a_n) < q.$$

4. From δ_i select the one that has the shortest base. If there are more than one such decomposition, from among them select the one that has the highest degree. If there are more than one such decomposition, unify them to get a single decomposition.

Actually, the step 3 above is required only if all the alternative decompositions obtained in the previous step are have low occurrence counts, since in step 4 we prefer the decomposition that has a shorter base.

Stage 6. Final suffix and suffix-sequence sets

D_3 is the final decomposition set obtained from the input corpus. It is the lexicon that may be used for morphological analysis of any other text. The set of suffix-sequences, Q in D_3, is the final set of suffix-sequences obtained, and the set of suffix parts, S_2 in Q is the set of suffixes obtained.

When the above process is run over the newspaper corpus A, the results can be summarised as:

Total number of words in the input corpus	:	116096
Number of distinct words in the input corpus	:	20140
Number of entries in the lexicon D_3	:	15707
Number of bases in the lexicon	:	10203
Number of morphological extension parts in S_2	:	428
Actual number of suffixes present	:	187
Precision of suffix identification	:	65.71%
Recall of suffix identification	:	73.80%
f-measure	:	69.52 %
Number of suffix sequences in Q	:	810.

When the exercise is carried out over a corpus of 301271 words (**corpus B**) from 525 news articles, in the initial suffix list S_1 we have,

Number of entries in the initial suffix list S_1	:	500
No. of valid suffixes, s	:	136
No. of compound parts, c	:	89
No. of composite suffixes, q	:	188
No. of invalid morphological extensions, b	:	87
Actual no. of suffixes present, n	:	190
Precision, $(s/(s+b))$	:	60.99%
Recall, (s/n)	:	71.58%
f-measure	:	65.86%.

The final lexicon obtained from the corpus B can be briefly summarised as:

Total number of words in the input corpus	:	301271
Number of distinct words in the input corpus	:	34559
Number of entries in the lexicon D_3	:	26509
No. of words that can be extracted from the lexicon	:	39098
Number of bases in the lexicon	:	15094
Number of entries in final suffix list S_2	:	381
No. of valid suffixes, s	:	136
No. of compound parts	:	102
No. of composite suffixes	:	76
No. of invalid morphological extensions, b	:	67
Number of suffix sequences in Q	:	1741
Actual number of suffixes present, n	:	190
Precision of suffix identification $(s/(s+b))$	:	67.00%
Recall of suffix identification (s/n)	:	71.58%
f-measure	:	69.21%.

The number of suffix-sequences is large compared to the number of individual suffixes, and the occurrence counts of most of these are low. This implies the possibility that there can be valid suffix-sequences other than those we have included in Q. In section 5.7 we discuss an approach to deal with suffix-sequences not encountered during training.

Like the suffix-sequences, the D_3 may not provide the complete decompositions for some words. This is because, the decomposition of each word depends on the presence of its other related words. Words whose sufficient number of other related forms have not occurred may be left incompletely decomposed. Hence, in a subsequent morphological analysis exercise of a test text, the decompositions in D_3 may be re-analysed taking into consideration the evidence from the test input.

4.16 Summary

In this chapter we have looked at some existing methods for unsupervised acquisition of morphology from a text corpus, and seen that the results from these methods leave scope for improvement. The issues involved can be broadly put in three categories– inherent issues of corpus based techniques, language specific issues, and issues due to the script and its usage for a particular language. The prominent issues inherent in corpus based techniques are– presence of *noise* in the form of non-words, foreign words and abbreviations, sparseness of some features, and ambiguity. Some language specific issues are– suffixes occurring detached from the base, presence of composite suffixes (suffix-sequences), and presence of compounds in addition to suffixed words. Also, in case of certain words there is ambiguity as to whether they are suffixed words or compounds. Since morphology manifests primarily in the spoken form of words, its acquisition from a text corpus depends on how faithfully the phonological content of the words reflected by the script used. In this respect the effectiveness of different scripts and their usage for a particular language is different. We have discussed a series of steps that takes care of most of the issues that arise in an attempt of unsupervised morphology acquisition for Assamese.

After initial experiments with an Assamese corpus of about 1,16,000 words, through which we define the steps required to effectively acquire the morphology of the language, we finally take a corpus of about 3,00,000 words for training using the steps developed. Through this training we define the set of suffixes in the language, as well as, build a *morphological lexicon* of the language. These

deliverables serve as computationally suitable representation of the morphological elements in evidence in the corpus and the way these elements combine to form words of the corpus. In the subsequent chapters we use this representation for morphological analysis of test input and to carry out further analysis of the evidence.

(Table 4.14 of suffixes in the corpus B:)

/dwy দ্বয়	/sth স্থ	/sthit স্থিত
/znY জ্ঞ	A া (আ)	A/ntr ান্তর (আন্তর)
A/nwit ান্বিত (আন্বিত)	A/tmk াত্মক (আত্মক)	Ab াব (আব)
Ami ামি (আমি)	AwH াৱহ (আৱহ)	Ay ায় (আয়)
Ayn ায়ন (আয়ন)	H*eten হেঁতেন	H*k হঁক
H*t হঁত	HIn হীন	He হে
Hi হি	I ী (ঈ)	It ীত (ঈত)
Iy ীয় (ঈয়)	IyA ীয়া (ঈয়া)	N ণ
Ni ণি	TA টা	Ti টি
To টো	Xm ক্ষম	ai ই
ao ও	ao* ওঁ	aowA ওৱা
b ব	b/ddh বদ্ধ	bA বা
bAd বাদ	bAr বাৰ	bHul বহুল
bRh/nd বৃন্দ	bîg বর্গ	bhAwe ভাৱে
bhi/ttik ভিত্তিক	bhu/kt ভুক্ত	bi বি
biHIn বিহীন	bid বিদ	bilAk বিলাক
bor বোৰ	brN বৰণ	cAm চাম
ch ছ	chA ছা	che ছে
chil ছিল	cho ছো	chowA চোৱা
con চোন	dAr দাৰ	dhrNe ধৰণে
dhrNr ধৰণৰ	di দি	e ে (এ)
ere এৰে	gE গৈ	gN গণ
g_r/st গ্রস্ত	grAkI গৰাকী	gt গত
i ি(হ্ব)	ib ইব	ik ইক
ikA ইকা	il ইল	ile ইলে
ilo ইলো	im ইম	it ইত
itA ইতা	je যে	jogJ যোগ্য
joge যোগে	ju/kt যুক্ত	k ক
k/lpe কল্পে	kAmI কামী	kAr কাৰ
kArI কাৰী	kArk কাৰক	kE কৈ
kN কণ	kRht কৃত	k_rme ক্রমে
keiTA কেইটা	*keibidh কেইবিধ*	*keidin কেইদিন*

keigrAkI কেইগৰাকী	keikhn কেইখন	keimAH কেইমাহ
keizn কেইজন	khini খিনি	khn খন
khni খনি	kn কন	kr কৰ
krN কৰণ	l ল	lA লা
lE লৈ	le লে	lgA লগা
lgIyA লগীয়া	li লি	lo লো
lok লোক	lowA লোৱা	m ম
mAn মান	mUlk মূলক	mîme মৰ্মে
mokorA মোকোৱা	mte মতে	muThi মুঠি
mukhI মুখী	muwA মুৱা	my ময়
n ন	nA না	nI নী
nIy নীয়	ne নে	ni নি
no নো	o ও	o* ওঁ
ok ওক	otA ওতা	owA ওৱা
pUîN পূর্ণ	pîj/nt পর্যন্ত	p_rA/pt প্রাপ্ত
p_rd প্রদ	pr পৰ	prA পৰা
pu/ST পুষ্ট	r ৰ	rAzi ৰাজি
rUp ৰূপ	rUpe ৰূপে	re ৰে
rt ৰত	s/mp/nn সম্পন্ন	sH সহ
sHite সহিতে	sHkAre সহকাৰে	sUck সূচক
shAlI শালী	shIl শীল	skl সকল
smUH সমূহ	t ত	tA তা
tIyA তীয়া	te তে	tm তম
to তো	tr তৰ	/tw ৎ
uowA উওৱা	uwA উৱা	wAn ৱান
wîtI ৱর্তী	we ৱে	y য়
yA য়া	ye য়ে	yk য়ক
yo য়ো	zA*i জাঁই	zAt জাত
zAtIy জাতীয়	zI জী	ze জে
zn জন	znA জনা	znI জনী
znk জনক	zopA জোপা	zorA জোৰা
zuri জুৰি		

Table 4.14: Suffixes in the corpus B (of about 300000 words)

Chapter 5

Morphological Analysis of Words in a Text

5.1 Introduction

In the process of figuring out the meaning of a natural language expression, morphological analysis is one of the first steps. Morphological analysis facilitates *recognition* of words that appear in expressions, since if the composition (or structure) of a word matches some known morphological framework, then several attributes of the word can be guessed. The result of morphological analysis is immediately useful in syntax analysis of expressions, and also in subsequent semantic analysis. For a language where concatenative morphological phenomenon is extensive, the problem of morphological analysis is primarily that of determining the decompositions that give the root words and additional morphemes that make up each word in a text. In a way, this is just what was attempted in Chapter 4. The important difference is that in there our main objective was to identify the morphological extensions, and in this chapter we try to identify the roots of the words in an input text. The method described in Chapter 4 requires a large input corpus for the purpose, whereas, the general need is to carry out morphological analysis for much smaller pieces of texts. So in the overall scheme, the exercise described in Chapter 4 is a *training phase*, during which enough knowledge is accumulated in the system in the form of

a set of suffixes and suffix-sequences and a lexicon. This knowledge is then used for morphological analysis of words in new text passages. In this chapter, we address the general problem of morphological analysis using the *resources obtained through the previous exercise.*

5.2 Word stemming

The problem of morphological analysis is commonly studied as the *word stemming* problem in the following problem context: given a text of a language and a list of suffixes in the language, decompose the words in the corpus into roots and suffixes wherever applicable. The first step towards this task may be simply to check the applicability of each suffix in each word, as is done in the well known Porter's method([36]). Breaking up of word by simply suffix-matching is likely to result in several incorrect decompositions too. For instance, *sender=send+er* is a correct decomposition whereas *gender=gend+er* is not. Correctness of a decomposition means that (1) the root identified by stripping a suffix from a given word is a valid word, *and* (2) the given word is actually derived from the identified root by applying that suffix. In Porter's and other similar methods ([36, 39]), this problem is addressed by specifying criteria for applicability of the suffixes. These criteria are based on the structure (spelling pattern) of the base part, and are manually formulated. For instance, the rule $SSES \rightarrow SS$ means strip suffix ES if the root ends in SS. Similarly, the rule $(*v*)ING \rightarrow (NULL)$ means strip suffix ING if the root contains a vowel. Such rules prohibit indiscriminate stripping of suffixes from word endings, and are found to be effective in preventing many invalid decompositions. In Assamese a noticeable feature is the extensive inflection of nouns, including proper nouns. In such words there are hardly any patterns that would facilitate specification of criteria like those given above. Moreover, it requires careful study of the morphology of the language to define such criteria, whereas, we would like the system to carry out stemming with minimum direct linguistic input.

In our unsupervised approach, we have relied on gathering support from *similar* words during decomposition of words, unlike the Porter's method where

each input word is considered in isolation for applicability of the suffixes. But this means that in our method decomposition of a word is possible only if there are adequate number of similar words in the input. This may be too big a demand while processing small or moderate sized texts. Hence to achieve effectiveness applying the method, we fall back upon the lexicon that was built in the previous exercise. This *lexicon* is maintained in a format that would facilitate easy morphological analysis of texts. In this chapter, we discuss a method of morphological analysis that uses the lexicon as well as the occurrence of possibly related words in the input text, to obtain the analyses of the words. We also discuss an approach to handle words for whose decomposition, adequate support is not found in the training corpus as well as in the input text.

5.3 Morphological analysis of new texts

For the words of a given text, morphological analysis can be carried out by identifying decompositions using the available suffixes and suffix-sequences. Hence, a word w_s in the input text may be decomposed as

$$\delta : [w = b + x]_s,$$

where x_s is a known suffix or suffix sequence. This decomposition is valid if the base b_s is valid. For example, of the following decompositions

1. $[mAnuHzn = mAnuH + zn]_A$ (মানুহজন) /the person/
2. $[p\_ryozn = p\_ryo + zn]_A$ (প্ৰয়োজন) /need/,

the first one is valid and the second is not. There is no word p_ryo_A, and p_ryozn_A is a root word. If an exhaustive lexicon is available, the occurrence of the base in the lexicon can be taken as the sufficient condition for validity of the base. When a lexicon is not available or we do not find the base in the lexicon, we consider the set of words in the input text, as an additional source of information. Still, for some valid decomposition, we may not find the base either in the lexicon or as a word in the input text. We refer to such a base as a *fresh base*. For a fresh base, we rely on the fact that in a highly inflectional language like Assamese, most root words have more than one inflected form. For instance, there are several inflected forms of the root $mAnuH_A$, such as $mAnuHr_A$ (/of human/), $mAnuHk_A$ (/human,

accusative/), $mAnuHbor_{A}$ (/the people/), and so on. Hence, corresponding to an inflected word in an input text, other inflected forms of a root too are likely to occur in the input text. The decompositions of these words would also involve the same base. So, even if we do not have direct evidence of the base either in the lexicon or in the input text, we can accept it as valid if it is involved in more than a threshold, t, number of decompositions of input words. The value of t can be chosen as 2 or more. In general, for a word w_s in the input text the decomposition δ as shown above is considered as valid if

1. the base b_s can be extracted from an available lexicon, or

2. there are at least t ($t >= 2$) words, $w_{i=1..t}$, in the input text for which we can obtain the decompositions

$$[w_i = b + x_i]_s, \quad i = 1, ..., t$$

where, x_i is a known suffix or suffix sequence, or $NULL$.

If for a particular base b_s, there are s possible distinct decompositions for input words or words extracted from the lexicon, we say that the base is *supported by* the s decompositions. Alternatively, we say the *support* for the base b_s is s. Insistence on good support improves the precision of the process. Usually, a minimum threshold value 2 for support is adequate. A higher threshold may cause valid bases to be discarded.

Considering suffix-sequences along with single suffixes while figuring out decompositions of input words is useful since it can provide the essential support value for some bases. If s_1, s_2 and s_3 are suffixes and s_2s_3 is a suffix-sequence, and if the words bs_1 and bs_2s_3 occur in the corpus, the support for the base b_s is 2. For, example, due to the decompositions

1. $[mAnuHzn = mAnuH + zn]_{A}$ (মানুহজন) /the person/
2. $[mAnuHborr = mAnuH + bor + r]_{A}$ (মানুহবোৰৰ) /of the people/

the support for the base $mAnuH_{A}$ is 2. Here, the second decomposition is possible because of a suffix-sequence.

Shortcomings of *support*

Support of the base of a decomposition is essentially an indication of whether there is evidence of the base being *part* of some other word too. Though the support gives a good estimation of the validity of a base as well as that of the decomposition, there is room for other criteria too, for improving the performance of the exercise. Broadly, we have to consider the following situations-

1. There can be some valid base words which take very few suffixes, or in a given input text a base might occur with an inadequate number of different suffixes For example, it is desirable to recognize the word $teishTA_{A}$ as

   $[teishTA = teish + TA]_{A}$ (তেইশটা) /twenty-three number of/,

   but the base, which is a number written in words, is unlikely to occur with any other suffix (except $NULL$) in a given text passage, and is declared as invalid. Discarding such decompositions reduces the recall of the exercise. In section 5.8 we discuss an approach to deal with such bases.

2. A decomposition involving a base with adequate support may also be invalid. It is possible that the base is invalid despite its adequate support. For example, the following two decompositions are invalid:

   $[colA = co + lA]_{A}$ (চোলা) /shirt/
   $[cor = co + r]_{A}$ (চোর) /thief/.

   The above decompositions are invalid despite the support for the base being 2. The base co_{A} is an invalid base. In general it is difficult to entirely prevent such decompositions from being selected. Still, to reduce the possibility of such decompositions we can adopt the following criteria-

   > if the base of an decomposition is shorter than a threshold, say 2 phonemes, then compute support for it within the given input text.

   The idea behind this criteria is that short letter strings are comparatively less stable semantically (see section 4.8.2) and may appear as the leading portion of unrelated words too. On the other hand, if the base of a

decomposition has occurred as part of some other word within that same discourse, the words are more likely to be have been derived from that base. Hence to offset the uncertainty associated with a short base its support is computed within the same discourse. Accordingly, the two invalid decompositions in the example above will be rejected unless the two words $colA_{\mathcal{A}}$ (/shirt/), and $cor_{\mathcal{A}}$ (/thief/) occur in the same input text.

3. A decomposition involving a valid base can also be invalid. For example, the decomposition-

$$[HAtI = HAt + I]_{\mathcal{A}} \qquad \text{হাতী} = \text{হাত} + \text{ঈ}$$

is invalid even though $HAt_{\mathcal{A}}$ is a valid base. The base is occurring as the leading part of two unrelated words– $HAt_{\mathcal{A}}$ and $HAtI_{\mathcal{A}}$. According to our argument in point 2, this may happen with short bases, and the selection criteria specified above would filter out most such cases. The effectiveness of the criteria depends on the threshold length of the base for considering it as a short base. In the example above, the base has two phonemes, and hence escapes that criteria. Increasing the threshold causes some valid decompositions to be discarded, and if the two words occur in the same discourse, the filtering criteria will be ineffective since the base will have the requisite support. In short, the possibility of producing such erroneous can only be reduced, not eliminated.

5.4 Decomposition evidence from the lexicon

It is important to take into account the nature of the decomposition evidence that the lexicon created through the process described in the last chapter (refer to section 4.15), provides. The following are some deficiencies of the lexicon:

1. Some words that can be extracted from the lexicon are not independent words in the language. For example, due to the two words $AlocnA_{\mathcal{A}}$ (আলোচনা /discussion/), and $Alocit_{\mathcal{A}}$ (আলোচিত /discussed (*participle*)/), in the training corpus, and the suffixes $nA_{\mathcal{A}}$ and $it_{\mathcal{A}}$ we can have the lexicon entries-

$$< AlocnA \quad Aloc{+}nA >_{A}$$
$$< Alocit \quad Aloc{+}it >_{A}$$

though the base $Aloc_{A}$ is not actually a valid word in Assamese.

2. Some decompositions in the lexicon are *incomplete*, *eg.*, in the decomposition

   $[AgreprA = Agr + e + prA]_{A}$ (আগৱেপৰা) /since the past/

   the base can be further decomposed as

   $[Agr = Ag + r]_{A}$ (আগৰ) /of past/ .

   This happens because the extent to which a word is decomposed depends on the presence of different words derived from the same root, and also on whether the suffix-sequence in a deeper decomposition has adequate occurrence count during the training phase.

3. Some decompositions in the lexicon are invalid because the suffix list used in building the lexicon contains few invalid suffixes too. For example,

   $[u\_tsAH = u\_ts + AH]_{A}$ (উৎসাহ) /encouragement/.

4. Some decompositions in the lexicon are invalid though they involve valid bases and morphological extensions. For example, the following decomposition in the lexicon is invalid since HAt_{A} means "hand" and $HAtI_{A}$ means "elephant":

   $[HAtI = HAt + I]_{A}$ ([হাতী = হাত + ঈ]) .

5.5 Multiple decompositions for a word

For figuring out the possible morphological analyses of the input words, we use the set of suffixes as well as suffix-sequences. So, if an input word has a suffix sequence in it, it will be possible to decompose it with only the final suffix in it, and also with one or more suffix-sequences. For example, for the word $gAlhHiHe*tenne_{A}$ (গালিহিহেঁতেননে /would (you) have come and sung (*inquisition*)/) the following decompositions can be figured:

1. $[gAliHiHe*tenne = gAliHiHe*tenne + NULL]_A$
2. $[gAliHiHe*tenne = gAliHiHe*ten + ne]_A$
3. $[gAliHiHe*tenne = gAliHi + He*ten + ne]_A$
4. $[gAliHiHe*tenne = gAli + Hi + He*ten + ne]_A$
5. $[gAliHiHe*tenne = gA + li + Hi + He*ten + ne]_A$.

Trivial decompositions, such as 1 above, are always valid, but they are useful only when no valid higher degree decompositions are possible. From the possible non-trivial decompositions we have to select the one that is valid and identifies the largest number of constituent parts in the word. In other words, we seek the decomposition that gives the highest *precision* and *recall*.

In [43], it was suggested that a good precision can be obtained by selecting the decomposition with the highest support for the base and longest length of base.

5.5.1 Context in decompositions

In the previous example all the candidate analyses shown are valid decompositions of the given word. It may also happen that of the multiple decompositions figured for a word some are invalid. For example, the second decomposition shown below for the word $bipdznk_A$ (বিপদজনক /dangerous/) is not valid:

1. $[bipdznk = bipd + znk]_A$
2. $[bipdznk = bipd + zn + k]_A$.

An useful criterion for selecting an analysis from multiple possible morphological analyses of a word is to take into account the *context* in decompositions. The context can provide clues for the selection of the correct decomposition. To understand this, consider the following two valid decompositions that are produced during the training phase:

1. $[AmodznkbhAwe = Amod + znk + bhAwe]_A$
   ([আমোদ + জনক + ভাবে]) /amusingly/
2. $[mAnuHznk = mAnuH + zn + k]_A$
   ([মানুহ + জন + ক]) /the person (*accusative*)/.

These decompositions imply that zn_{A}, znk_{A}, k_{A}, and $bhAwe_{A}$ are suffixes, and $(znk + bhAwe)_{A}$ and $(zn + k)_{A}$ are two valid suffix-sequences. Further, in the training phase we do not come across any decomposition that involves the suffix sequence $(zn + k + bhAwe)_{A}$ since no single root word in Assamese takes the suffixes zn_{A}, znk_{A} and $znkbhAwe_{A}$. Now, if the test input contains the words $shixkznk_{A}$ (শিক্ষকজনক /the teacher (*accusative*)/), $bipdznk_{A}$ (বিপদজনক /dangerous/), and $bipdznkbhAwe_{A}$ (বিপদজনকভাৱে /dangerously/), we have the following analyses (or decompositions) for them using the available suffixes and suffix-sequences:

1. $[shixkznk = shixk + znk]_{A}$
2. $[shixkznk = shixk + zn + k]_{A}$
3. $[bipdznk = bipd + znk]_{A}$
4. $[bipdznk = bipd + zn + k]_{A}$
5. $[bipdznkbhAwe = bipdznk + bhAwe]_{A}$
6. $[bipdznkbhAwe = bipd + znk + bhAwe]_{A}$.

For the word $shixkznk_{A}$ decomposition 2 should be retained, while for $bipdznk_{A}$ decomposition 3 should be retained. For $bipdznk_{A}$ decomposition 4 is not selected because it is not consistent with the analysis of the word that can be extracted from the decomposition 6. However, if the word $bipdznkbhAwe_{A}$ is not there in the input, we would select decomposition 4 since it has a higher degree than decomposition 3. Even the lexicon might contain analysis 4 for the word $bipdznk_{A}$ if the training corpus contains the words $bipd_{A}$ and $bipdznk_{A}$ but not $bipdznkbhAwe_{A}$.

To generalize the above criteria, let us call the a decomposition of a word a *non-terminal decomposition* if it can be extracted from a longer available decomposition. If no longer decomposition is available we call it a *terminal decomposition*. For example, if the input text contains the word $bipdznkbhAwe_{A}$, decomposition 3 for $bipdznk_{A}$ is a non-terminal decomposition, and decomposition 4 is a terminal decomposition since it cannot be extracted from any other longer decomposition. The required criterion is then, *in the presence of a non-terminal decomposition, a different terminal decomposition for a word should be discarded.* In other words, if a word in the input can be obtained

by adding a morphological extension to a shorter word, then the shorter word should not be decomposed in a way that it cannot be extended to obtain the longer word. This follows the intuition that the longer word provides a stronger *context* for decomposition of the shorter word. The important implication of this criteria is that, even a decomposition available in the lexicon may be discarded if the words in a given input text so requires.

5.6 Steps for morphological analysis

In the light of the above discussion, we now summarize the steps for morphological analysis. *Though for most cases, the choice of analyses for a given set of words may be made through simple steps, in some cases a careful consideration of the input conditions as well as prior knowledge of the lexicon, is necessary.* We refer to the set of words in the input text as T, and the set of suffixes and composite suffixes (*i.e.*, concatenated suffix-sequences) as S.

Stage 1. Produce decompositions relating different input words

We assume that the input is a *coherent text* so that words with similar initial letter-strings are derived from the same base if the differing trailing portions match known suffixes or suffix-sequences. So we identify decompositions

$$\delta : [w = b + x_1 + ... + x_n]_S$$

such that, $x_{i=1\ n} \in (S \cup \{NULL\})$, b_S has a support greater than 1, and each of the word longer than b_S that can be extracted from δ is in the input. That is, $bx_1...x_i \in T$, $1 <= i <= n$. To obtain such decompositions, the steps may be-

1. Identify decompositions, $[w = b+x]_S$, where, $w_S \in T$, $x_S \in S$, and support of b_S is greater than 1.
2. *Recursively reduce* the bases of the decompositions (see section 4.11.1). If a decomposition of a word is *included* in some other decomposition of that word, drop that decomposition.
3. Perform *compaction* of the decompositions (see section 4.11.4).

Let us refer to the set of decompositions so obtained as D_1. We note that some

morphological extension parts in D_1 may be composite suffixes from S, and must be broken up eventually.

Stage 2. Find lexicon entries for decompositions in D_1

The evidence available in the input text limits the degree of the decompositions in D_1. For some of these decompositions higher degree decompositions can be actually possible. For example, if the input text contains the words, $rA/ST\_rIy_{\mathcal{A}}$ (রাষ্ট্রীয় /national/) and $rA/ST\_rIytAbAd\dot{I}sklr_{\mathcal{A}}$ (রাষ্ট্রীয়তাবাদীসকলৰ /of the nationalists/) then we have the following decomposition in D_1:

$$\delta : [rA/ST\_rIytAbAdIsklr = rA/ST\_rIy + tAbAdIsklr]_{\mathcal{A}} .$$

We consider two cases of available lexicon entries:

Case 1: The lexicon contains the decomposition

δ_{l1} :

$$[rA/ST\_rIytAbAdIsklrHe = rA/ST\_r + Iy + tA + bAdI + skl + r + He]_{\mathcal{A}}$$

(রাষ্ট্রীয়তাবাদীসকলৰহে) /of the nationalists rather/

which contains all the partition points present in the decomposition δ. In this situation, we take the relevant portion of the decomposition in the lexicon, *i.e.*,

$$[rA/ST\_rIytAbAdIsklr = rA/ST\_r + Iy + tA + bAdI + skl + r]_{\mathcal{A}} .$$

Case 2: Instead of δ_{l1} the lexicon contains the decomposition

$$\delta_{l2} : [rA/ST\_rIytAbAdIsklr = rA/ST\_rIytA + bAdI + skl + r]_{\mathcal{A}} .$$

Here, δ_{l2} is shallower (*i.e.*, it has a longer base) than δ, but its morphological extension portion contains all the partition points present in the *corresponding portion* of δ. Hence we take the relevant portion of the decomposition δ_{l2} and *unify* it with the decomposition δ (see section 4.11.4) to obtain the decomposition

$$[rA/ST\_rIytAbAdIsklr = rA/ST\_rIy + tA + bAdI + skl + r]_{\mathcal{A}} .$$

Let us refer to the set of decompositions we obtain by the above steps as D_2. Recall that D_1 may contain more than one distinct decomposition for some words. So D_2 may also contain more than one distinct decomposition for some words. The reason why we seek lexicon entries only after forming the decompositions in

D_1 is that we want to take into account the longest *context* available in the input for the shorter words. As explained in section 5.5.1, lexicon entries for shorter words might be incorrect.

Stage 3. Words not decomposed in D_2

The set D_2 provides analyses of words for which evidence is available in the lexicon. For input words for which no non-trivial decomposition is provided by D_2, we revert back to D_1 and look for decompositions in it. Recall that in D_1 there may be morphological extension parts that are composite suffixes (see section), which should be broken up. Suppose such a decomposition in D_1 is

$$\delta : [w = b + x_1 + ... + x_n]_s$$

where one or more of the $(x_{i=1..n})_s$ are composite suffixes. If there exist any alternative suffix sequence for $(x_1+...+x_n)_s$ which contains all its partition points, using them we obtain all the alternate decompositions for w_s (see section 4.11.2). Otherwise, for these decompositions we obtain new alternative suffix sequence as described in section 5.7. Though D_2 does not contain decompositions for these words, some decompositions in D_2 may match the leading portions of these words. We refer to a pair of distinct words as *siblings* if one can be extracted from the decomposition of the other or, their decompositions have one or more common partition points. From the alternative decompositions, we select the ones with longest sibling match with some decomposition in D_2. If there are more than one such decompositions, we select the one that has a degree not higher than the others. We add the selected decompositions in the set D_2.

Stage 4. Root words and compound decompositions

For those words for which no non-trivial decomposition is found, we try to identify compound decompositions, that is, decompose into two parts both of which can be extracted from the lexicon or D_1. From among the undecomposed words left, the ones for which no decomposition using the given set of suffixes is possible, irrespective of the support of the base involved, are confirmed roots. Those words for which decompositions are possible but the bases involved have very poor support (*i.e.*, base has not occurred in any other decomposition), we consider the number of occurrence of the word. If the word has occurred several

times, say more than 10 times, that word may be considered as root. In section 5.8 we discuss this in little more detail.

Summary of above the steps

The important underlying assumption in the above analysis exercise is that the input text is a single coherent discourse such that if one input word can be obtained by appending a known suffix or suffix-sequence to another input word, then the two words are actually related. In the steps in section we put together related words to form longest decompositions possible. These long decompositions provide *context-evidence*, which can help in avoiding invalid decomposition. The decompositions that we obtain may have scope for further break-up. In stage 2 we seek relevant evidence from the lexicon to further analyse the input words represented in the decompositions. In stage 3 we deal with the words for which suitable decomposition evidence is not found in the lexicon. Some of the words that are left undecomposed after this may be compounds. In stage 4 we attempt to recognize compounds that are formed from other known words. From the words that are still left unanalysed, we declare as roots the words that have occurred several times in the input.

5.7 New suffix-sequences

As pointed out in page 82, though the given set of suffixes can be almost exhaustive, the set of suffix-sequences may not be so. During analysis of words in a test passage we may encounter *new suffix-sequences*. Consider the decomposition δ of section -

$$\delta : [w = b + x_1 + ... + x_n]_S \ .$$

Let x_S denote the parts-sequence $(x_1 + ... + x_n)_S$. If x_S is not a known suffix-sequence, but there are known alternative suffix-sequences of x_S that contain all the partition points of the latter, w_S is decomposed using such suffix-sequences. Otherwise, x_S is a new parts-sequence, which we assume to be valid, since each individual part in x_S is valid and δ is obtained from words that we assume are related. To obtain suitable suffix-sequences from x_S we first replace the composite

suffix parts in it with their respective equivalent suffix-sequences. Some composite suffixes can have more than one equivalent suffix-sequence (*eg.*, the composite suffix znk_S has two equivalent suffix-sequences– znk_S and $(zn+k)_S$). Hence for x_S we may obtain more than one alternative suffix-sequences, none of which is already known. For example, suppose we have the decomposition

$$\delta: \quad [bipdznkbhAweHe = bipd + znk + bhAweHe]_A$$
(বিপদজনকভাবেহে) /rather dangerously/,

where the parts-sequence $(znk+bhAweHe)_A$ is not a known suffix-sequence. The parts znk_A and $bhAweHe_A$ are composite suffixes. From δ we then obtain the decompositions

1. $[bipdznkbhAweHe = bipd + znk + bhAwe + He]_A$
2. $[bipdznkbhAweHe = bipd + zn + k + bhAwe + He]_A$,

involving new suffix-sequences. The second decomposition above is invalid since the suffix-sequence $(zn + k + bhAwe + He)_A$ is invalid. To determine whether a suffix-sequence x_S is valid we verify whether each 2-part sub-sequence in x_S is a known suffix sequence. If each 2-part sub-sequence is known, the suffix-sequence x_S is valid, otherwise it is invalid.

In general, to verify if the suffix-sequence $x_S : (x_1 + ... + x_n)_S$ is valid, we verify each *l-part* subsequence of x_S

$$(x_1 + ... + x_l)_S,\ (x_2 + ... + x_{l+1})_S,\ ...,\ (x_{n-l+1} + ... + x_n)_S.$$

If each such sub-sequence is a known suffix-sequence, or has adequate support in the analysis generated for the given input text, then x_S is valid.

5.8 Decomposition involving base with poor support

At the end of the analysis of the words we have some words for which some decomposition is possible but the support for the bases in them is very poor. A close examination shows that the likelihood of such decompositions being valid is related to the number of times these words have occurred in the input, *i.e.*, the frequency of the words. If in a text a word has high frequency, it roughly

implies that the word is *prominent* in that discourse. If the decomposition of a prominent word is valid, then it means that the base of the decomposition is also semantically prominent in that discourse. In a highly inflectional language, a prominent base is likely to occur with multiple distinct suffixes. That means the support of the base should be good. Conversely, if the support for a supposedly prominent base is low, the base is probably invalid. So, we can say that if the support of the base involved in the decomposition of a word is low despite a high frequency of the word, that decomposition discarded as invalid. On the other hand, if the support for a base is low, *i.e.*, it has occurred the decomposition of very few (say, only one) words, but those words too have occurred only a small number of times in the input (inadequate evidence), it is not clear whether the base (and the decomposition) are invalid.

We observe the above phenomenon in the experimental analysis of a moderate sized corpus of about 49000 words. In this experiment we consider the frequency of words for which the possible decompositions involve bases with *support* 1. The results are summarized in Table 5.1. We observe that among the decompositions with smaller base frequencies there are more valid decompositions compared to those with larger base frequencies. Also, fewer decompositions have very high frequencies of base and more have small frequencies of base. We present a small example from the experiment:

Example

In the decomposition

$$[kumAr = kumA + r]_{\mathcal{A}} \quad (\text{কুমার=কুমা+র}) \quad /\text{boy}/$$

$kumA_{\mathcal{A}}$, which is not a valid base has a support value only 1, but its frequency is 74 (since $kumAr$ occurs 74 times). Similarly, in the decomposition

$$[kthA = kth + A]_{\mathcal{A}} \quad (\text{কথা=কথ+া}) \quad /\text{matter(conveyed)}/$$

$kth_{\mathcal{A}}$ is not a valid base and its support is 1 and frequency is 160. On the other hand, in the decomposition

$$[AgDokhrte = AgDokhr + te]_{\mathcal{A}} \quad (\text{আগডোখরতে=আগডোখর+তে})$$

/at the front portion/

the base $AgDokhr_{\mathcal{A}}$ is actually a valid base despite the support being 1. We find that its frequency is 1.

Base Freq	No. of decompositions	No. of valid decompositions	Precision (%)
1	3081	1068	34 66
2	599	142	23.70
3	292	46	15.75
4	158	16	10.12
5	95	9	9.47
6	66	4	6.06
7	56	9	16.07
8	30	2	6.66
9	26	1	3.84
10	25	4	16.00
11	24	1	4.16
12-191	168	14	8.33

Table 5.1: Quality of decompositions with low (=1) base support

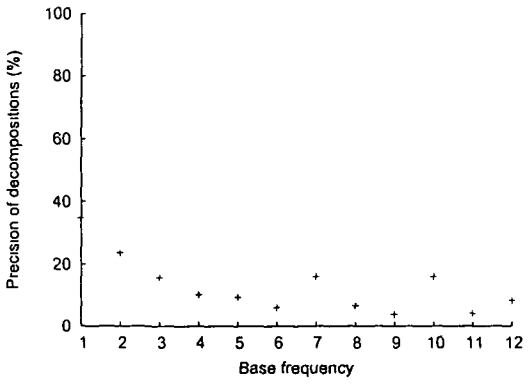


Figure 5.1: Precision of decompositions with low (=1) base support

5.9 Measuring the quality of morphological analysis

In our morphological analysis exercise, on one hand, we strive to identify as many morphemes as possible in the input words, and on the other, we exercise caution

to avoid identifying wrong morphemes. However, it is difficult to quantify the performance of the exercise in terms of the parts identified. For example, the ideal analysis of the word $l'rAborklE_A$ (ল'ৰাবোৰকলৈ /with the boys/) is

$$[l'rA + bor + k + lE]_A.$$

If the computational method produces the analysis

$$[l'rAbor + klE]_A,$$

neither of the two parts is actually ideal. So both precision and recall would be 0%, though it is clear that the *partition point* (see section 4.5) identified is valid. Hence, to quantify the performance of this exercise in terms of *precision* and *recall*, we count the partition points identified in the words, and compare it with the number of partition points that should ideally be identified in the words. To account for the undecomposed words we consider the ends as partition points, and refer to them as *trivial partition points*. Each trivial partition point is a valid partition point. Thus, *recall*, which denotes the ratio of the number of valid partition points identified to the number of partition points to be ideally identified, can be computed as

$$recall = \frac{V + C}{A + R} \tag{5.1}$$

where, V is the number of *valid* non-trivial partition points identified, C is the number of undecomposed words that are actually root words (each presents a trivial partition point), A is the total number of non-trivial partition points to be ideally identified, and R is the number of root words present (each presents a trivial partition point).

Precision denotes the ratio of the number of valid cases identified to the total number of cases identified. In our exercise, we can compute this as

$$precision = \frac{V + U}{I + U} \tag{5.2}$$

where, U is the number of undecomposed words (each presents a *valid* trivial partition point), and I is the total number of non-trivial partition points identified. Alternatively, if the trivial partition points in words which should ideally have been decomposed, are treated as invalid, we can compute precision as

$$precision = \frac{V + C}{I + U} \quad . \tag{5.3}$$

The numerator in equation 5.2 is greater than (or equal to) that of equation 5.3 since in the former if a word that should have been decomposed is left undecomposed, it is treated as a "missed" partition point, and not an invalid partition point. As a case of "missed" partition point, it is accounted for in the recall value.

Thus, for the analysis $[l'rAborklE = l'rAbor + klE]_A$ recall is 33% and precision is 100%. For the analysis $[l'rAbor = l'rAbor]_A$, the recall is 0%, the precision according to equation 5.2 is 100%, and according to equation 5.3 is 0%.

In a text some words occur multiple times and in our morphological analysis method, we consider only the distinct words in the input. Hence to get the actual and identified partition point counts of all the words in the text, we multiply the counts associated with each distinct word by their respective occurrence counts. For instance, if in a corpus of n distinct words, word w_i occurs m_i, and the number of non-trivial partition points identified in it is p_i, then

$$I = \sum_{i=1}^{n} p_i * m_i \quad .$$

5.10 Results of morphological analysis experiment

We have tested the morphological analysis approach outlined in section 5.6 over text chunks from different sources. We have used the lexicon and set of suffixes and suffix-sequences obtained through the process described in chapter 4 from the **corpus B** of about 301271 words (see page 81). Corpus B is a collection of 525 newspaper articles that include general news, sports news and editorial articles. For testing we have run our process over 84 other newspaper articles totalling 32271 words from the same newspaper source, and 66 articles from the Emille corpus for Assamese (http.//www.ling.lancs.ac.uk/corplang/emille/zipfiles/assamese.zip) totalling 138131 words. The Emille corpus articles used for testing are from various domains, namely, agriculture, anthropology, astrology, astronomy, biographies, business, industry, media, music, novels, stories, translated literature, travel, *etc.*

First we observe the effectiveness of our process quantitatively. For this, for each input text we take the following counts:

- Number of words in the input text that are in the training corpus too.
- Number of words in the input text that can be obtained from the lexicon already produced from the training corpus.
- Number of words in the input text that can be obtained through the morphological analysis method discussed in section 5.6.

Morphologically analysed words are not necessarily decomposed words. They are words whose structures have been decided through the analysis process. Unmatched words, *i.e.*, the words that are not analysed, are words which have not occurred in the training corpus, and have occurred very few number of times (mostly only once) in the test passages. Many of these words are actually root words and hence do not require any decomposition.

The quantitative results of the experiment are summarized in Tables 5.2 and 5.3, and depicted graphically in figures 5.2 and 5.3.

To evaluate the result of our morphological analysis tests qualitatively, we have to verify the analysis produced for each word in the input as described in section 5.9. Since this requires intensive manual effort, we have chosen to perform the verification for samples taken from the analysis results. We have taken samples of different types of test inputs, *viz.*, different types of newspaper articles, and different types of articles from the Emille corpus. For the words of a particular input text, we manually prepare the correct analyses of all the words in that text, and compare the counts and appropriateness of the partition points with the those produced by our morphological analysis method (see 5.9). More specifically, we compute the following:

- Total words in the input file (Column T in Table 5.4)
- Total partition points generated (Column A in Table 5.4)
- Spurious partition points generated (Column B in Table 5.4)
- Total undecomposed words (Column C in Table 5.4)
- Actual partition points required (Column D in Table 5.4)

Recognition* (%)	100	99	98	97	96	95	94	93	92	91	90
No of texts	0	0	2	7	10	11	8	11	10	4	4

Recognition* (%)	89	88	87	86	85	84	83	82	81	80	
No of texts	8	3	1	2	1	1	0	0	0	1	

(a) *Distribution of test input words' recognition percentage from training words*

Recognition* (%)	100	99	98	97	96	95	94	93	92	91	90
No of texts	0	0	5	7	10	10	7	15	5	7	4

Recognition* (%)	89	88	87	86	85	84	83	82	81	80	
No of texts	8	0	3	1	0	1	0	0	0	1	

(b) *Distribution of test input words' recognition percentage from lexicon*

Recognition* (%)	100	99	98	97	96	95	94	93	92
No of texts	4	24	35	15	3	2	0	0	1

(c) *Distribution of test input words' recognition through our morphological analysis*

* Fractional part of percentage values are truncated

Table 5 2 Word recognition performance for 84 newspaper articles

Recognition* (%)	83	82	81	80	79	78	77	76	75	74
No of texts	1	2	4	4	2	4	8	3	3	2

Recognition* (%)	73	72	71	70	69	68	67	66	65	64
No of texts	6	3	6	5	2	2	2	1	1	2

Recognition* (%)	63	62	61	60	59	58	57	56	55	
No of texts	1	1	0	0	0	0	0	0	1	

(a) *Distribution of test input words' recognition percentage from training words.*

Recognition* (%)	83	82	81	80	79	78	77	76	75	74	73
No of texts	1	2	7	2	6	6	4	4	1	6	5

Recognition* (%)	72	71	70	69	68	67	66	65	64	57	
No of texts	4	1	6	1	4	1	1	2	1	1	

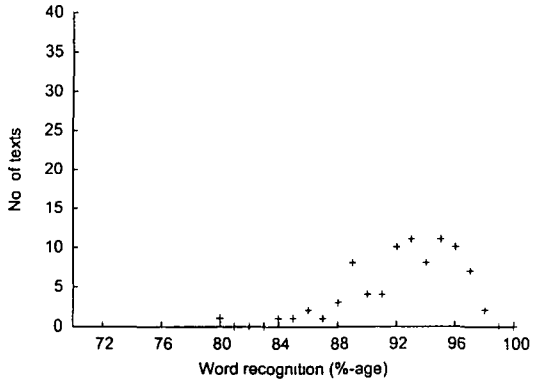
(b) *Distribution of test input words' recognition percentage from lexicon.*

Recognition* (%)	97	96	95	94	93	92	91	90	88	87
No of texts	3	3	11	14	13	11	5	4	1	1

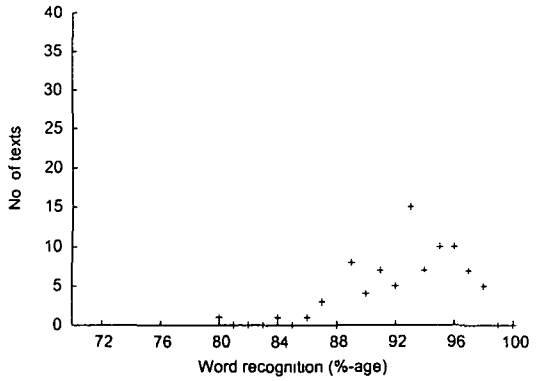
(c) *Distribution of test input words' recognition through our morphological analysis.*

* Fractional part of percentage values are truncated.

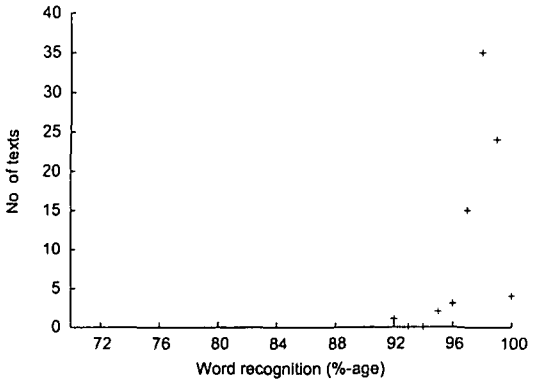
Table 5.3: Word recognition performance for 66 Emille corpus articles



(a) *Distribution of test input words' recognition percentage from training words*

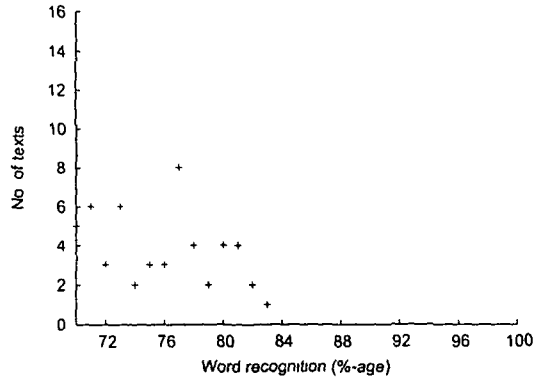


(b) *Distribution of test input words' recognition percentage from lexicon*

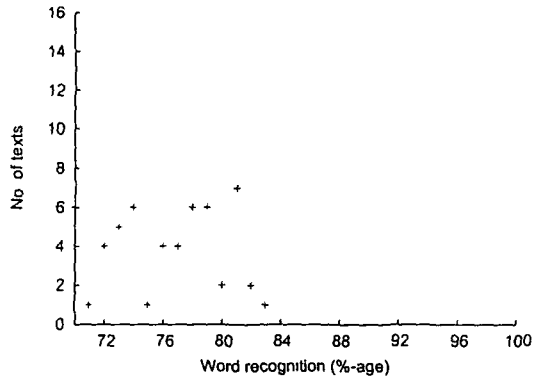


(c) *Distribution of test input words' recognition through our morphological analysis.*

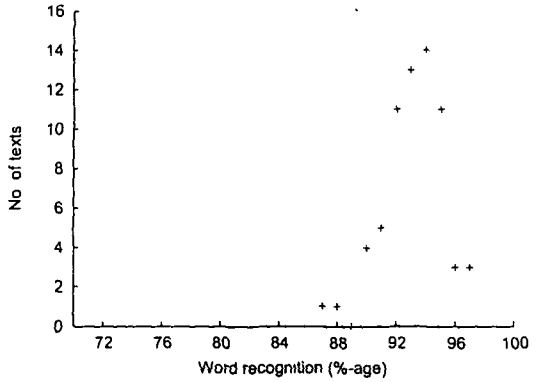
Figure 5 2 Word recognition performance for 84 newspaper articles



(a) *Distribution of test input words' recognition percentage from training words.*



(b) *Distribution of test input words' recognition percentage from lexicon.*



(c) *Distribution of test input words' recognition through our morphological analysis.*

Figure 5.3: Word recognition performance for 66 Emille corpus articles

- Actual roots (Column E in Table 5.4)
- Valid partition points missed (Column F in Table 5.4)
- Undecomposed words that are roots (Column G in Table 5.4)
- Precision, P_1 (equation 5.2) ((A+C-B)/(A+C) in Table 5.4)
- Precision, P_2 (equation 5.3) ((A+G-B)/(A+C) in Table 5.4)
- Recall (equation 5.1) ((A+G-B)/(D+E) in Table 5.4)

Table 5.5 shows a small sample input text and the morphological analysis produced by our method as well as manually. As mentioned at the beginning of this chapter, other methods such as the ones by Gaussier ([16]) and Goldsmith ([19]) work with large input corpus. On the other hand, methods such as Porter's ([36]) use hand coded rules. The nature of the problem we tackle is thus, distinct. So we do not compare the result of those methods with ours.

5.11 Summary

In this chapter we have described an approach for morphological analysis of text of a highly inflectional language. We have presented intermediate results as well as final results of such morphological analysis trials for Assamese texts. The final results show that on an average we have been able to carry out the task of morphological analysis of Assamese texts with precision of around 90% and recall of around 85% (Table 5.4). This is significant, because our entire approach has been an unsupervised one. From one point of view we have achieved what has been our primary goal. In this chapter we have assumed that the input text that we process is correct in terms of spellings and morphology. In the next chapter we move on to, what we would call, a "higher" level of morphological processing, *i.e.*, word classification based on their morphological behaviour. We opine that the results of the classification can be used to take care of morphologically incorrect words to some extent.

Text id	T	A	B	C	D	E	F	G	Precision		Recall
									P_1	P_2	
1	350	197	36	198	181	196	20	181	90 89	86 58	90 72
2	615	358	38	331	366	317	46	301	94 48	90 13	90 92
3	468	246	30	263	262	248	46	232	94 11	88 02	87 84
4	213	109	17	122	101	123	9	114	92 64	89 18	91 96
5	351	228	20	173	242	165	34	157	95 01	91 02	89 68
6	419	296	47	207	296	203	47	182	90 66	85 69	86 37
7	292	190	34	153	180	147	24	135	90 09	84 84	88 99
8	770	438	63	395	426	403	51	360	92 44	88 24	88 66
9	792	437	65	441	416	437	44	412	92 60	89 29	91 91
10	514	322	36	271	356	245	70	234	93 93	87 69	86 52

(a) Evaluation for newspaper articles

Text		T	A	B	C	D	E	F	G	Precision		Recall
Id	Type									P_1	P_2	
1	Mu	2193	1177	251	1273	1248	1187	322	1056	89 76	80 90	81 40
2	Mu	2079	1072	198	1175	1154	1063	280	946	91 19	81 00	82 09
3	Cm	1644	871	141	971	887	924	157	860	92 35	86 32	87 80
4	Cm	1734	902	189	1004	907	1000	194	889	90 08	84 05	84 01
5	As	2037	1036	205	1190	1080	1124	249	1019	90 79	83 11	83 94
6	As	1893	948	204	1092	985	1044	241	936	90 00	82 35	82 80
7	Bg	1961	1095	244	1119	1121	1052	270	940	88 98	80 89	82 42
8	Bg	2049	1150	243	1182	1277	1036	370	944	89 58	79 37	80 03
9	Nv	2395	1418	349	1290	1610	1136	541	974	87 11	75 44	74 40
10	Nv	2423	1695	394	1184	1743	1079	442	922	86 31	77 21	78 77
11	St	2334	1311	263	1313	1528	1096	480	999	89 98	78 01	78 01
12	St	2391	1511	290	1229	1737	1055	516	945	89 42	79 05	77 58
13	Tr	3041	1657	298	1660	1758	1555	399	1386	91 02	82 76	82 86
14	Tr	2889	1550	242	1652	1711	1526	403	1409	92 44	84 85	83 94
15	Md	1901	1122	199	1033	1204	974	281	880	90 77	83 67	82 78
16	Md	1516	845	218	822	783	850	156	713	86 92	80 38	82 06
17	Ot	3133	1719	314	1731	1873	1567	468	1410	90 90	81 59	81 83
18	Ot	1320	680	115	733	730	678	165	619	91 86	83 79	84 09
19	Ag	1271	700	151	724	729	669	180	603	89 40	80 90	82 40
20	An	2427	1244	239	1439	1394	1280	389	1174	91 09	81 22	81 49
21	An	2493	1261	201	1457	1444	1321	384	1231	92 60	84 29	82 86
22	An	2273	1129	178	1272	1190	1200	239	1095	92 59	85 21	85 61
23	An	2328	1101	234	1374	1139	1326	272	1188	90 55	83 03	83 37
24	Bs	2369	1482	172	1263	1632	1177	322	1081	93 73	87 10	85 12
25	TL	2086	1121	211	1199	1233	1085	323	987	90 91	81 77	81 84
26	TL	1400	826	198	776	853	712	225	656	87 64	80 15	82 04
27	TL	1650	954	192	906	1040	809	278	743	89 68	80 91	81 40

(b) Evaluation for Emille corpus articles

Columns

T	Total input words	A	Total non-trivial partition points identified
B	Spurious partition points identified	C	Total undecomposed words
D	Actual non-trivial partition points required	E	Actual roots
F	Valid non-trivial partition points missed	G	Correct root recognitions
P_1	Precision using equation 5 2	P_2	Precision using equation 5 3
Recall	using equation 5 1		

Text types

Mu	Music	Cm	Commerce	As	Astrology	Bg	Biography
Nv	Novel	St	Story	Tr	Travel	Md	Media
Ot	Other	Ag	Agriculture	An	Astronomy	An	Anthropology
Bs	Business	TL	Translation literature				

Table 5 4 Evaluation of morphological analysis

(a) Portion of original text:

eıXet_rt ek udAHrN dAnGı dhrı asm zmıytr mukhpAt_rgrAkIye ky- '/sthAyI bAsı/ndAr p_rmANpt_r p_rdAnr bıSyToweı cAok | rAzJ crkAre /sthAyI bAsı/ndAr p_rmANpt_r p_rdAn p_rk_rıyA b/ndh rAkhıche | crkArr eı sı/ddhA/ntaı ponpTIyAbhAwe XtısAdhn krıche s#khJAlghu znsAdhArNk | kArN /sthAyI bAsı/ndAr p_rmANpt_r noHowA Hetu ArXI, af̂dhsAmrık bAHınI Adıt nıyu/ktı powA bhAles#khJk s#khJAlghu juwke cAkrıt jogdAn krıb prA nAı | /sthAyI bAsı/ndAr p_rmANpt_r punr p_rdAnr bAbe rAzJ crkArk p_ryozn mAt_r eTA kebıneT sı/ddhA/ntr | athc crkArkhne AzılEke eı sı/ddhA/nt /g_rHN krıb nAı | eyA mAt_r eTA sru udAHrNHe | enedhrNr bHu udAHrN dıb pArı jAr /dwArA rAzJr k#/g_rechI crkArkhne s#khJAlghu znsAdhArNr smsJAk gurutw nıdıyAr bıSyTo p_rtIymAn Hy | Aclte k#/g_rechI crkArkhne AmAk bı/shwAst l'b prA nAı |'

(b) Analysis by our method:

eı+Xet_rt ek udAHrN dAnGı dhr+ı asm zmıyt+r mukhpAt_r+grAkI+ye ky- '/sthAyI bAsı/ndA+r p_rmAN+pt_r p_rdAn+r bıSyTo+weı cAok | rAzJ crkAr+e /sthAyI bAsı/ndA+r p_rmAN+pt_r p_rdAn p_rk_rıyA b/ndh rAkh+ı+ch+e | crkAr+r eı sı/ddhA/nt+aı ponpTIyA+bhAwe XtısAdhn krı+che s#khJAlghu znsAdhArN+k | kAr+N /sthAyI bAsı/ndA+r p_rmAN+pt_r noHowA Hetu ArXI, af̂dhsAmrık bAHınI Adıt nıyu/kt+ı powA bhAles#khJk s#khJAlghu juw+k+e cAkrı+t jog+dAn krı+b prA nAı | /sthAyI bAsı/ndA+r p_rmAN+pt_r punr p_rdAn+r bAbe rAzJ crkAr+k p_ryozn mAt_r eTA kebıneT sı/ddhA/nt+r | athc crkAr+khn+e Azı+lE+k+e eı sı/ddhA/nt /g_rHN krı+b nAı | eyA mAt_r eTA sru udAHrN+He | ene+dhrN+r bHu udAHrN dıb pAr+ı jAr /dwAr+A rAz+J+r k#/g_rech+I crkAr+khn+e s#khJAlghu znsAdhArN+r smsJA+k gurutw nıdıyA+r bıSyTo p_rtIymAn Hy | Acl+t+e k#/g_rech+I crkAr+khn+e AmAk bı/shw+As+t l'b prA nAı |'

(c) Manual analysis:

eı+Xet_rt ek udAHrN dAnGı dhr+ı asm zmıyt+r mukhpAt_r+grAkI+ye ky- '/sthAyI bAsı/ndA+r p_rmAN+pt_r p_rdAn+r bıSy+To+we+ı cA+ok | rAzJ crkAr+e /sthAyI bAsı/ndA+r p_rmAN+pt_r p_rdAn p_rk_rıyA b/ndh rAkh+ı+ch+e | crkAr+r eı sı/ddhA/nt+aı ponpTIyA+bhAwe XtısAdhn kr+ı+che s#khJAlghu znsAdhArN+k | kArN /sthAyI bAsı/ndA+r p_rmAN+pt_r noHowA Hetu ArXI, af̂dhsAmrık bAHınI Adı+t nıyu/kt+ı powA bhAles#khJk s#khJAlghu juwk+e cAkrı+t jog+dAn krı+b prA nAı | /sthAyI bAsı/ndA+r p_rmAN+pt_r punr p_rdAn+r bAbe rAzJ crkAr+k p_ryozn mAt_r eTA kebıneT sı/ddhA/nt+r | athc crkAr+khn+e Azı+lE+ke eı sı/ddhA/nt /g_rHN krı+b nAı | eyA mAt_r eTA sru udAHrN+He | ene+dhrN+r bHu udAHrN dı+b pAr+ı jAr /dwArA rAzJ+r k#/g_rech+I crkAr+khn+e s#khJAlghu znsAdhArN+r smsJA+k gurutw nıdı+yA+r bıSy+To p_rtIymAn Hy | Acl+t+e k#/g_rech+I crkAr+khn+e AmAk bı/shwAs+t l'b prA nAı |'

Table 5 5 Sample morphological analysis of an input text portion

Chapter 6

Classification of Words

6.1 Introduction

In the process of figuring out the meaning of a natural language expression morphological analysis is one of the first steps. The immediate aim of morphological analysis is to facilitate syntax analysis in which the words in the expression are grouped as phrases and then successively into some known structures. This grouping of words is done not on the basis of the exact meanings of the words, but on the basis of certain attributes of the words. Several words of the language may share the same attributes and hence their individual roles in sentence formation are identical. Thus words are generally classified into categories, which govern their role in sentence formation. Traditionally words are classified as *nouns, verbs, adjectives, adverbs, conjunction, etc.* However, for proper syntax analysis of sentences more precise classification is often done. For instance, there are proper nouns, countable nouns, *etc.* Verbs may be transitive, non-transitive or bi-transitive. In addition, in morphologically rich languages some other attributes, such as number, tense, *etc.*, may also be reflected by modification of the word forms.

Though a human user of a language may not perform explicit classification of words into categories, there is no doubt that in a formal linguistic analysis, such as syntax or semantic analysis (*eg.*, [1]), the knowledge of the category of each word is important. This knowledge is useful in part-of-speech (POS)

tagging of texts. In traditional linguistic exercises, the categories of the words are determined manually using different sources of information – from existing catalogs (*dictionaries*) to context of usage of each word. In computational linguistics, the task is not simple because often the required catalogs are not available in a computationally useful form, and the mechanisms to analyse the context of the usage of the words are not adequate. Most of the reported work in POS tagging are based on the use of some existing computational lexicon and a pre-defined *tag-set*. However, there are languages, such as Assamese, for which no suitable computational lexicon that can provide the class information of words is available. Our objective is to *develop* a lexicon where the classes of the words are indicated, using the evidence available in a text corpus. To this end, in the preceding chapters, we have discussed methods to identify the suffixes in the language from a raw text corpus, and then to decompose words in a given text into base and suffix-sequences. Using these methods we developed a *morphological lexicon* of Assamese from a text corpus (page 81). In this chapter we discuss some computational methods that consider the evidence of affix usage in the training corpus to identify the underlying classes of words in the language. The classes identified are not exactly POS classes (*eg.*, [29]) used for tagging words in a text; POS classes are more *syntactic*. The classes we identify determine the morphological behaviour of the words. The morphological behaviour depends on, apart from the potential syntactic roles of the word, other factors such as

- the phonetic structure of the word. For instance, the ergative case marker in Assamese depends on the way the base is pronounced, *eg.*,

$[gruwe = gru + we]_{\text{A}}$	(গৰুৱে)	/cow (*ergative*)/
$[crAiye = crAi + ye]_{\text{A}}$	(চৰাইয়ে)	/bird (*ergative*)/
$[kAchai = kAch + ai]_{\text{A}}$	(কাছই)	/tortoise (*ergative*)/
$[mAche = mAch + e]_{\text{A}}$	(মাছে)	/fish (*ergative*)/.

- empirical criteria. For instance, the choice of suffix as determiners for nouns in Assamese is sometimes empirical, *eg.*,

$[ndIkhn = ndI + khn]_{\text{A}}$	(নদীখন)	/the river/
$[rA/stATo = rA/stA + To]_{\text{A}}$	(ৰাস্তাটো)	/the road/.

The category we identify for a word can be included as an attribute of the word in the lexicon. Though we have carried out our experiments for Assamese, but we strongly feel that the methods are general and applicable for most inflectional languages.

6.2 Word sense and classification

Linguistic categories of words, such as noun, verb, adjective, *etc.*, reflect meaning of the words, although to a very limited extent. For example, if a word is categorized as a noun, it implies that the word denotes an *object*, either physical or abstract. From one perspective, the entire meanings of words can be viewed as successive classification, though classification beyond a point is not explicitly done. For example,

pen : *category*: noun, physical;
usage: writing;
dimension: 10-15 cm long; (relevant to *physical*)
inscription material: ink; (relevant to *writing*)
etc.

This might seem to imply that classification of words requires the knowledge of the meanings of the words. Many computational methods take as input a pre-determined list of word categories or parts-of-speech, and pre-specified criteria to classify the words into these categories (*eg.*, [5, 46]). *Is it possible to classify words when no prior information about either the meanings of the words, or the underlying categories of words, is available?* This is often a problem in computational linguistics.

The solution to the computational problem given above probably lies in the analysis of the structure of the words and their usage in sentences or phrases. Either or both of these display some *patterns* that can help in categorizing the words to a certain extent, though the final meaning may not be inferred from the structure of the word or the sentence alone. Different approaches have been taken by researchers that make use of such observations as well as related statistics for word classification. (*eg.* [5, 6, 21, 46, 38, 31, 45]). These approaches generally adopt one of the two broad approaches - rule-based and stochastic. Further,

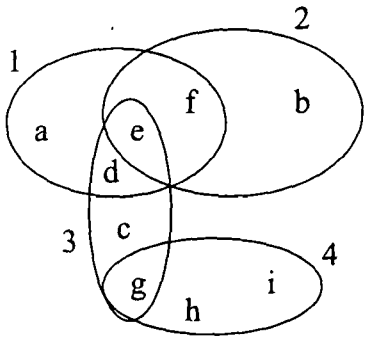
they either use unsupervised or supervised training. However, most such work target part-of-speech identification, rather than a general context-independent categorization of the words.

In this chapter, we discuss identification of underlying categories of base words in a language and then the classification of words into these categories by considering the morphology, and more specifically, the use of suffixes in the words. Unlike many other approaches, we do not consider the context (neighbouring words) for each word, but the suffixed (inflected) forms of each base in a corpus, to guess the category of the base word. A suffixed word can be seen as a sequence of the morphemes, and thus an *n-gram*. We consider multiple available n-grams where the base is the same, to guess the category of the inflected word or the base morpheme. The category identified for a root word is expected to be a useful piece of information and can be recorded against the word in a lexicon.

6.3 Classification of words by suffix evidence

Each suffix in a language affects a base word making it suitable to play a *specific role* in a sentence. For example, in the English word *play-ed*, the suffix *-ed* makes the base word *play* suitable to represent an action in the past. It is also understandable that such an effect by a suffix can be expected for words of a certain type of meaning only. In the above example, the suffix *-ed* applies to words representing action, *i.e.*, *verbs*. By considering the application of suffixes to words carefully, we can guess the categories of base words. Of course, there can be ambiguities too in this. For example, the suffix *-s* in English applies to verbs and nouns with different roles. In this discussion our notion of linguistic category is– *two words are in the same category if the set of suffixes that they can take is the same*. Accordingly we try to identify the set (or group) of suffixes corresponding to each category of the base words. For a hypothetical language L, the grouping of suffixes by the category of base words can be depicted pictorially as in Figure 6.1. The suffixes *c, d, e, f, g* do not clearly imply any distinct category for the words that take these suffixes, and the suffixes *h, i* do not indicate different categories. On the other hand, the suffixes *a, b, c, h, i* unambiguously imply the

categories 1, 2, 3, 4, 4 respectively.



The letters $a, b, c, d, e, f, g, h, i$ denote different suffixes, and the ellipses numbered 1, 2, 3, and 4, enclosing the letters denote linguistic categories of words that occur with those suffixes.

Figure 6.1: Suffixes and linguistic categories of words for language L

In a highly inflectional language like Assamese, the word categories identified by suffix applicability can be more precise than the usual linguistic categories, such as noun, verb, *etc.* That is, say, within nouns we may be able to find subclasses. For instance, in Assamese different determiners are allowed with different categories of nouns. The idea of such categorization of words is considered in [37] for the purpose of *morpho-syntactic parsing.* However, input and output of the system described there is different from those of our method. In a computational linguistic exercise, it is often the case that the input to the system is only a text corpus, and possibly, with a break-up of words into base and suffix wherever applicable. Definitions of the word categories in terms of suffixes are not given to the system. The problem is to identify the underlying linguistic categories of words (*1,2,3* and *4* in Figure 6.1) in the language and classify the words into these categories.

Identification of *patterns* of affixation in words in a corpus, termed as *signatures*, has been described in [19]. Words are grouped according to such signatures invoking the principles of Minimum Description Length (MDL) framework. Since each individual word in a corpus do not always occur with all affixes valid for it, such groupings based on direct evidence from a corpus may not hold beyond that corpus. What is required is some further analysis of the

affixes.

As in the case of acquisition of the suffixes described in Chapter 4 and morphological analysis of words in Chapter 5, in our approach we attempt to identify the word categories in the language from a sufficiently large training corpus, and then carry out classification of words of a given text. In suffix-based classification of base words, the set of suffixes that are seen with the base word is of prime significance. We call the set of suffixes seen with a word its *suffix characteristic*, or simply *characteristic* of the word. Some simple approaches for categorising words in a corpus according to their suffix characteristics have been mentioned in [43]. We discuss them briefly in sections 6.3.1 thru' 6.3.3. In the experiments reported in these sections, we have considered the suffixation evidence in the lexicon obtained from the training corpus B. Since the lexicon is obtained by our unsupervised morphology acquisition approach (see page 81), hence the suffixation evidence in it is not completely valid– there are some invalid suffixes as well as invalid decompositions in it. To minimize such evidence during classification, we have considered only those suffixes, which have occurred in at least t different decompositions. The value of t chosen depends on the size of the corpus. Again, to make the evidence "richer" in terms of number of suffixation cases, we have considered suffixed words as bases when such words have further suffixes added to them (cases of suffix-sequences). For example, if the lexicon contains the decomposition – $[sbhA + khn + r]_{\mathcal{A}}$, we count two bases, *viz.*, $sbhA_{\mathcal{A}}$ and $sbhAkhn_{\mathcal{A}}$ with the suffixes $khn_{\mathcal{A}}$ and $r_{\mathcal{A}}$ respectively. Briefly, the evidence has the following dimensions–

Size of the training corpus	: 301271 words (corpus B)
Threshold frequency of suffixes, t	: 12
Number of suffixes with adequate frequency	: 122
Number of distinct bases with selected suffixes	: 12530.

6.3.1 Direct classification based on characteristics

The simplest idea for classification of words is to form groups of words with exactly matching suffix characteristics. However, this leads to too many classes of words, because in a corpus many words are likely to occur only with a subset of

the set of linguistically valid suffixes for it. Different words of the same linguistic category can occur with different subsets of suffixes in the input. Hence their characteristics are different and they are classified into different categories. In an experiment of this idea using the suffixation evidence described above, we obtain–

Number of categories of words identified : 1987.

6.3.2 Identifying subsets of characteristics

One attempt to overcome the drawback of the direct classification method is to assume that at least some words from each true linguistic category will occur with all or almost all valid suffixes for that category. We call such a suffix characteristic a *master characteristic*. The characteristics of all words which are of the same linguistic category will be subsets of a master characteristic. For example, suppose the suffix characteristic C_w, of word w_s contains all possible suffixes for the linguistic category of w_s, and the suffix characteristic of w_1 is a subset of C_w. Then we classify w_s and w_1 into the same class. In our experiment we obtain–

Number of categories of words identified : 641.

The number of word categories identified is large in this case too, and many of the categories are superfluous. This is mainly because, in most cases the suffix characteristics we consider as masters are actually subsets of the exhaustive suffix sets of real word categories.

6.3.3 Merging overlapping characteristics

The drawback of the idea of subsets described above is that for a linguistic category, hardly any word occurs with all valid suffixes for that class. To overcome this we modify the idea and compute a *synthesized master characteristic* of each linguistic category by taking *union* of its tentative subsets. A pair of tentative subsets of a *synthesized master* are identified as two characteristics that have at least k common elements. We call this synthesized master characteristic, a *closure*, where k is the *degree of closure*. We start by selecting the largest of all characteristics, and assume that it is the closure. Then sequentially for each

remaining characteristic, C, we determine if C has at least k elements common with the closure or C has less than k elements, which are all common with the closure. If so, we update the closure by taking its union with C. If during one *pass* of such testing of characteristics the closure actually gets updated, we perform another pass considering the characteristics that failed the test in the previous pass(es). This continues till the closure is not updated in a particular pass. Then, we proceed to generate another closure by starting with the largest characteristic from among the ones not included in the previous closures. Higher degree of closure leads to more categories to be identified. In our experiment using the evidence described above, the results can be summarized as:

Closure degree (k)	No. of categories of words
2	5
3	5
4	18
5	32
6	41

The major drawback of the results obtained is that with each value of k, one of the categories obtained contains too many suffixes, including suffixes from distinct linguistic categories, whereas most of the remaining categories have too few suffixes. For example, with $k = 5$ the set of suffixes attributed to one of the categories is–

{ *zuri, zorA, zn, zJoti, yo, ye, y, we, wAn, w, u, tw, tA, t, smUH, skl, shIl, sh, sbhA, sH, s/mp/nn, s#khJk, rUp, rAz, r, pti, pt_r, p_rsAd, p_rA/pt, pr̂j/nt, pUr̂N, owA, o, ni, nAth, n, my, mukhI, mu/kt, mte, mr̂me, mUlk, mAn, m/nt_rI, m, lE, l, krN, khn, khini, keizn, keikhn, keiTA, k_rme, kE, kAr, kA, k, ju/kt, joge, it, inGr, ik, ichil, i, grAkI, ghr, gE, g, ere, e/shwr, e', e, din, dhrNe, dAn, ch, bor, blgIyA, birodhI, bilAk, bid, biHIn, bhi/ttik, bhAwe, bAsI, bAr, bAd, b, ao, aichil, aiche, ai, Xet_r, To, Ti, TA, N, J, IyA, Iy, I', I, Hi, He, HIn, H*t, H*eten, E, Ar, Al, A, /sth, /g_r/st, /dwy, #ı, # }$_S$,

and with $k = 2$ the set of suffixes attributed to one of the categories is–

$$\{\ yA,\ ch\ \}_S.$$

Neither of the above is a close approximation of the set of suffixes of a real category of words in the language. The larger set shown above contains suffixes, such as $H^*t_{\mathcal{A}}$ and $H^*eten_{\mathcal{A}}$, which do not ever occur with the same base. On the other hand, the smaller set shown above is inadequate; it should also include suffixes such as, $m_{\mathcal{A}}, ye_{\mathcal{A}}, blgIyA_{\mathcal{A}}, etc.$, that can be applied to the same bases. Using a large value of k, the size of the one large set of suffixes is reduced, but the number of *inadequate* small sets of suffixes increases.

From the results of the simple ideas in [43] it is seen that the problems in classification arise because of the following broad reasons:

- evidence provided by individual words in a corpus, is often sparse,
- suffix ambiguity– certain suffixes can be applied to words of different categories, and,
- word sense ambiguity– certain words actually belong to multiple linguistic categories. In Assamese word sense ambiguity is comparatively rare, though not altogether absent. For example, $kar_{\mathcal{A}}$ (কৰ) in one sense means "tax" (noun) and in another it means "do" (verb, imperative).

Before we discuss some approaches to address these issues we define a concept of *co-occurrence* of suffixes.

6.4 Co-occurrence of suffixes

Definition A suffix s_a is said to co-occur *with suffix s_b if there is at least one base word in the available evidence, which occur with both suffixes.*

For a suffix we define its *simple co-occurrence list* (or *simple co-occurrence set*), as the set of suffixes that co-occur with it. We denote the simple co-occurrence list of suffix σ as $C^s(\sigma)$. Let each suffix occur in its own simple co-occurrence list. That is,

$$\sigma \in C^s(\sigma).$$

If the corpus is large enough, we expect that the simple co-occurrence lists of all suffixes are exhaustive. For example, in case of the language L, the exhaustive

simple co-occurrence list of suffix a is { a, d, e, f }. This is not to say that there is necessarily some word in the corpus that occur with all the suffixes a, d, e and f. Again, some suffixes in a language are generally infrequent. For such suffixes the simple co-occurrence lists may not be exhaustive even if a large corpus is used. Hence, for classification purposes only those suffixes which have occurred more than a threshold, t, number of times in the training corpus, are considered. The value of t can be selected according to the size of the training corpus. In our experiments with a corpus of about 300000 words as input, we consider suffixes that have occurred at least 12 times (see page 118).

6.5 Suffix characteristic extension by co-occurrence

The major drawback of a *closure* obtained by merging overlapping characteristics (section 6.3.3) is that it contains suffixes which have not co-occurred with each other. To address this issue a possible approach is to take a suffix characteristic, C, and extend it by including all suffixes that have co-occurred with all the suffixes in C. Let the extended characteristic be C'. Using our notion of co-occurrence sets, C' can be obtained by taking an intersection of the simple co-occurrence sets of all the suffixes in C. That is,

$$\text{if} \quad C : \{\sigma_1, \sigma_2, ..., \sigma_n\}$$

$$\text{then} \quad C' = \bigcap_{i=1}^{n} C^s(\sigma_i).$$

That $C' \supseteq C$ can be seen from the fact that since C is a characteristic set, every suffix in C occurs with the base of which C is the characteristic set. That is, each suffix in C co-occurs with all other suffixes in C (due to that base). That is, the simple co-occurrence set of each suffix in C contains all the suffixes in C. Hence, the intersection of the simple co-occurrence sets of the suffixes in C will include all the suffixes in C. Now, suppose we have two characteristic sets C_1 and C_2 for bases b_1 and b_2 respectively, and $C_2 \subset C_1$. In simple words, this means that b_2 can belong to the same category as b_1. Since there is more suffixation

information about b_1 than we have about b_2, b_1 better describes the suffix set of that category Also, $C_2 \subset C_1$ and $C_1 \subseteq C_1'$, implies that $C_2 \subset C_1'$. Hence, once we compute C_1', the extended characteristic of C_1, we need not compute C_2' for C_2. Going a step further, if a characteristic set C is found to be the subset of an already computed *extended characteristic* (not simply a characteristic set), we do not compute the extended characteristic of C.

To implement this idea, we take up one by one the characteristic sets and compute the corresponding extended characteristic sets, provided the characteristic is not a subset of an already computed extended characteristic set. In doing so we take up the larger characteristic first.

Experimental results

When we tried the above method over the input evidence described in page 118, the results can be summarised as–

Number of categories of words identified : 246

Some of the extended characteristics are–

{ *zurı, zorA, t, smUH, sbhA, r, m/nt_rI, lE, l, khn, keıkhn, k, e, bor, bAsI, aı,* }$_A$
{ *t, smUH, r, pt_r, o, n, mte, mr̂me, lE, l, k_rme, kAr, k, joge, e, bor, ao, To, He,* }$_A$
{ *zn, y, t, skl, r, n, m, l, k, b, aichıl, aiche, aı, To, He,* }$_A$
{ *zn, y, tA, t, skl, rUp, r, n, mAn, lE, l, k, grAkI, e, bAd, ao, aı, To, I, He, A,* }$_A$
{ *zn, t, r, pUr̂N, o, mu/kt, k, e', e, To, I, HIn,* }$_A$.

The extended characteristics obtained are more realistic than the suffix sets obtained through the previous methods. However, there are still some groups of non-co-occurring suffixes in some of the extended characteristics. For example, the suffixes HIn_A and skl_A do not co-occur, but they figure together in some extended characteristic set. We discuss this issue section 6.7.

6.6 Pivot suffixes and word classification

An approach for word classification distinct from the ones discussed above is to consider *pivot suffixes* of word categories ([42]). We observe that in a language some suffixes apply to words of a distinct category, and others apply to words belonging to more than one category. For example, in English the suffix *ed* applies only to verbs, but the suffix *s* applies to verbs as well as nouns. Similarly, in Assamese the suffix $skl_{\mathcal{A}}$ applies only to nouns, but the suffix $e_{\mathcal{A}}$ applies to nouns as well as verbs. We define a *pivot suffix* as a suffix that applies only to words of a distinct linguistic category. Thus in English *ed* is a pivot suffix and *s* is not, and in Assamese $skl_{\mathcal{A}}$ is a pivot suffix and $e_{\mathcal{A}}$ is not. (In practice we find finer word categories for nouns in Assamese and $skl_{\mathcal{A}}$ may not be a pivot suffix then.) For language L of figure 6.1, *a*, *b*, *c* and *h* (or *i*) are pivot suffixes. So if a word occurs with any of these pivot suffixes, we can conclusively determine that the word belongs to the linguistic category represented by that pivot suffix. Now, our task is to identify the underlying word categories and the corresponding pivot suffixes.

Let us consider the situation depicted in figure 6.1. We observe that the pivot suffixes figure in only one linguistic category each, and the other occur in more than one category. We claim that among the suffixes of a given linguistic category, the pivot suffix(es) have the least number of co-occurring suffixes, and this number is exactly the number of suffixes in that linguistic category. For example, for the category denoted by the ellipse 1, the suffixes are a_S, e_S, d_S and f_S. The pivot suffix of this category is a_S and it has exactly these four suffixes in its co-occurrence list. This is true in general, because all non-pivot suffixes occur with at least one suffix from another category, in addition to the suffixes of the category that we are considering. Thus, among all the suffixes in the language, the one with the least number of co-occurring suffixes is a pivot suffix and it represents one word category. This implies that none of the other suffixes in the simple co-occurrence list of this suffix will be the pivot suffix of any other word category. To find a pivot suffix for another word category, we simply have to apply the same criteria (of lowest number of co-occurring suffixes) to the list of suffixes minus the already identified pivot suffixes and their co-occurring suffixes.

Proceeding like this, we can identify a pivot suffix for each word category. At any stage if we find more than one suffix having the least number of co-occurring suffixes, we may select any one of them as pivot and proceed as usual.

Each of the pivot suffixes identified by the above steps represents a distinct morphological category of words based on their suffixation behaviour. These word categories may not correspond to part-of-speech (POS) classes, which are related to the structure of a sentence rather than the morphology of the words. The co-occurrence list of a pivot suffix is the set of suffix applicable to the word category that it represents. A suffix that has the same simple co-occurrence list as that of a pivot suffix is considered a *co-pivot* of that pivot suffix. A pivot suffix and its co-pivot suffixes have the same significance in classification of words. Finally, each pivot suffix and the co-pivot suffixes occur in the simple co-occurrence list of only one pivot suffix. For the language of Figure 6.1 we identify a_s, b_s, c_s and either h_s or i_s as pivot suffixes.

Once the pivot suffixes, the co-pivot suffixes, and their simple co-occurrence lists are identified, the words in the training corpus or some other test corpus (where the words are already decomposed to reveal the presence of suffixes) can be classified into the categories corresponding to the pivot suffixes. In all the suffix based classification approaches, a base word can be put in a unique category only if it has occurred with adequate number of suffixes in the given evidence. Otherwise, there shall be more than one tentative categories for the word. In the pivot suffix based classification approach, if a base occurs with a pivot suffix or a co-pivot suffix, its classification will be *definite*, *i.e.*, the base can be classified into a unique category. Otherwise, the classification will be *tentative*, *i.e.*, more than a single category will be predicted for the base.

The steps for identification of pivot suffixes are summarised below:

1. Form a list of suffixes S in the training corpus that have occurred more than t (threshold) number of times, *i.e.*,
$$S = \{s_1, s_2, ...s_m\}_S \ .$$
2. For each suffix s_i such that $s_i \in S$, prepare the simple co-occurrence list, C_i. Note that $C_i \subset S$ and $s_i \in C_i$.
3. Sort the suffixes by non-decreasing number of elements in the simple co-

occurrence lists.

4. Mark the suffix that has the smallest simple co-occurrence list as the first pivot suffix p_1. Mark the other suffixes in its simple co-occurrence list as non-pivot suffixes.

5. Successively from the set of un-marked suffixes, mark the suffix with the smallest simple co-occurrence list as the next pivot suffix. Mark the other suffixes in its simple co-occurrence list as non-pivot suffixes.

6. For each suffix in the simple co-occurrence list of each pivot suffix, check if it occurs in the simple co-occurrence list of any other pivot suffix. If it does not, then mark it as a co-pivot of the pivot suffix.

The steps for classifying the base words of an input text into the categories represented by the pivot suffixes are summarised below:

1. If base b_s in the input word list W, has occurred with a pivot or co-pivot suffix, s_s, put b_s in the class represented by the pivot (or co-pivot) suffix s_s. This classification is definite. Else,

2. If for a base, b_s, with suffix characteristic C ($C \subset S$), no definite classification is possible, then tentatively put it in each class represented by a pivot suffix s_s, such that C is a subset of the simple co-occurrence list of s_s.

   This classification is tentative, *i.e.*, in the absence of any pivot suffix associated with the word, we simply predict that the word can eventually occur with the pivot suffix corresponding to any of the categories that we tentatively put the word in.

Complexity:

Suppose, from the training corpus we have m distinct decompositions involving n distinct suffixes that have occurred more than t (threshold) number of times in these decompositions. To obtain the simple co-occurrence lists of the n suffixes, first we group the decompositions by the bases. This can be done by sorting them on the bases with a computational complexity of $O(m \log m)$. If there are

g decompositions in a group (involving a particular base), the computational complexity of updating the simple co-occurrence lists of the g suffixes is $O(g^2)$. If there are k such groups, with the i^{th} group having g_i suffixes, total complexity is $\sum g_i^2$. Since $g_i \leq n$, the total computational complexity of obtaining the simple co-occurrence lists of all the n suffixes is $O(mn)$.

From the simple co-occurrence lists, to obtain the pivot suffixes, first we compute the sizes of the n simple co-occurrence lists. The total size of the lists is less than or equal to n^2. Hence the computational complexity of this task is $O(n^2)$. Then to sort the n suffixes in non-decreasing order of the sizes of their respective simple co-occurrence lists, the computational complexity is $O(n \log n)$. Finally, as we mark an un-marked suffix as a pivot suffix, we also mark the suffixes in its simple co-occurrence list as non-pivot suffixes. Hence, the complexity of this step is $O(n^2)$. Thus the overall computational complexity of finding the pivot suffixes is $O(m \log m + mn + n^2 + n \log n + n^2)$. Since it is expected that $m \gg n$ and $\log m < n$, so the asymptotic complexity is $O(mn)$.

The computational complexity of matching a suffix characteristic against another characteristic is $O(n)$. To classify a base word into one or more of the categories represented by the pivot suffixes, the characteristic of the word is matched against the simple co-occurrence lists of the pivot suffixes. Hence the computational complexity of the steps required is $O(pn)$, where p is the number of pivot suffixes.

6.6.1 Experimental results

We have tried the above method over the input described in page 118. Since the input is a morphological lexicon built with an unsupervised method, it contains invalid suffixes as well as invalid decompositions too. The results can be summarised as–

Number of pivot suffixes	: 26
Number of co-pivot suffixes	: 0
Number of definite classification of words	: 744.

The pivot suffixes identified are:

(#, /dwy, /g_r/st, Al, H*eten, H*t, I, aiche, bAr, bhAwe, bid, blgIyA, e, ere, ik, inGr, it, kA, k_rme, ke, keiTA, ne, owA, p_rsAd, s/mp/nn, yA)$_A$

(-ং, -দ্বয়, -গ্রস্ত, -াল, -হেঁতেন, -হঁত, -ী, -ইছে, -বাৰ, -ভাৱে, -বিদ, -বলগীয়া, ে-, ে-ৰে, ি-ক, ি-ঙৰ, ি-ত, -কা, -ক্রমে, -কে, -কেইটা, -নে, ে-ৱা, -প্রসাদ, -সম্পন্ন, -য়া)

The '-' marks in the suffixes listed in Assamese fonts indicate the position of the last letter of the base

The simple co-occurrence sets of the pivot suffixes are presented below (enclosed within braces { }) along with some example base words, and a rough criteria for applicability of the pivot suffix:

$\#_A$: {zn, t, smUH, sH, rAz, r, pti, o, ni, m, lE, khn, kAr, k, joge, i, ghr, e/shwr, e, bor, birodhI, bilAk, b, ao, ai, To, Ti, N, Iy, I, E, A, #}$_A$
Example bases: dl_A, kl_A.
Remark: $\#_A$ is an invalid suffix.

$/dwy_A$: {zn, yo, ye, y, tw, t, skl, sbhA, sH, rUp, r, o, lE, l, keizn, k, grAkI, ghr, e, bilAk, ao, ai, Iy, He, /dwy}$_A$
Example bases: $netA_A$, shi/lpI_A.
Remark: $/dwy_A$ is used with nouns (not proper nouns) indicating persons.

$/g\_r/st_A$: {y, t, smUH, r, o, n, my, mu/kt, kAr, k, ju/kt, i, ai, To, HIn, A, /g_r/st}$_A$
Example bases: $durnIti_A$, $At\#k_A$.
Remark: $/g\_r/st_A$ is used with nouns indicating conditions that prevail over some person, object or place.

Al_A : {zn, r, o, khn, grAkI, ei, e, TA, E, Al}$_A$.
Example bases: $bish_A$, $zIwnk_A$
Remark: Al_A is an invalid suffix.

H^*eten_A : {t, skl, r, prj/nt, o, n, lE, l, gE, e, ao, ai, To, Hi, He, H*eten}$_A$
Example bases: $thkA_A$, $pAle_A$.
Remark: H^*eten_A is used with verbs in past tense or participle forms.

H^*t_A : {*yo, y, w, skl, sH, r, o, l, k, grAkI, e, bor, ao, ai, Iy, He, H*t*}$_A$

Example bases: $shishu_A$, $juwtI_A$, $te/NDulkAr_A$.

Remark: H^*t_A is used with nouns indicating persons.

I'_A : {*t, smUH, sH, r, o, n, lE, l, khn, k, e/shwr, e, birodhI, ao, To, I', I, He*}$_A$

Example bases: $k\#/g\_rech_A$, $crkAr_A$.

Remark: The single-quote in I'_A is superfluous. This suffix is used with nouns that can be transformed into adjectives.

$aiche_A$: {*zn, y, t, skl, r, ni, n, m, l, kE, k, gE, b, aichil, aiche, ai, To, He*}$_A$

Example bases: $powA_A$, lgA_A.

Remark: $aiche_A$ is used with verb forms that end with a vowel, to convert them to present continuous forms.

bAr_A : {*zn, u, t, sh, s#khJk, r, o, mAn, l, khn, k, grAkI, e, din, dhrNe, bor, bAr, b, ao, ai, To, Ti, TA, N, A*}$_A$

Example bases: pA^*c_A, $tini_A$.

Remark: bAr_A is used with words denoting numbers.

$bhAwe_A$: {*zn, u, tw, tA, t, smUH, skl, s#khJk, rUp, rAz, r, o, n, mUlk, krN, khn, kE, kAr, k, i, e, dhrNe, bor, bhAwe, bAd, ao, ai, Xet_r, I, He, Ar, A*}$_A$

Example bases: $bhul_A$, $adhik_A$.

Remark: $bhAwe_A$ is used with words that denote some quality and can be used as nouns too.

bid_A : {*y, t, shIl, r, mukhI, mUlk, m/nt_rI, lE, l, k, g, dAn, bid, bhi/ttik, ao, ai, Xet_r, To, He*}$_A$

Example bases: $bhASA_A$, ci/ntA_A.

Remark: bid_A is used with words that indicate some subject or profession.

$blgIyA_A$: {*ye, y, t, skl, r, pUrN, mte, m, l, k, dhrNe, ch, blgIyA, b, ao, aichil, ai, To, N, He*}$_A$

Example bases: krA_A, znA_A.

Remark: $blgIyA_A$ is used with verbs in present tense.

e'_A : {*zn, w, u, t, smUH, sH, r, pt_r, pUrN, o, n, mu/kt, mte, lE, l, khn, k, ichıl, ı, e/shwr, e', e, dAn, bor, To, J, I, He, HIn, E, A, #i*}$_A$

Example bases: $mAn_{\mathcal{A}}$, $s\#kT_{\mathcal{A}}$.

Remark: The single-quote in $e'_{\mathcal{A}}$ is superfluous. This suffix is used with nouns (as ergative case marker) as well as verbs in the present imperfect forms. As such this is not an appropriate pivot suffix linguistically.

$ere_{\mathcal{A}}$: $\{t,\ smUH,\ r,\ p\_rA/pt,\ my,\ lE,\ khn,\ keikhn,\ kAr,\ k,\ ere,\ bor,\ A\}_{\mathcal{A}}$

Example bases: $s\#gIt_{\mathcal{A}}$, $nATk_{\mathcal{A}}$.

Remark: $ere_{\mathcal{A}}$ is used with nouns that can act as an "instrument" for some action.

$ik_{\mathcal{A}}$: $\{zJoti,\ t,\ smUH,\ r,\ pUr\hat{N},\ o,\ mukhI,\ m/nt\_rI,\ l,\ keikhn,\ k,\ joge,\ ik,\ i,\ e,\ dAn,\ bhi/ttik,\ ai,\ To,\ I,\ A\}_{\mathcal{A}}$

Example bases: $sAHitJ_{\mathcal{A}}$, $bJwsAy_{\mathcal{A}}$.

Remark: $ik_{\mathcal{A}}$ is used with nouns that denote some subject or profession (the suffix $bid_{\mathcal{A}}$ is not used with these nouns).

$inGr_{\mathcal{A}}$: $\{y,\ u,\ t,\ smUH,\ r,\ p\hat{r}j/nt,\ o,\ khini,\ k,\ inGr,\ ichil,\ i,\ e,\ dAn,\ ch,\ To,\ Ar,\ A,\ \#i\}_{\mathcal{A}}$

Example bases: $bl_{\mathcal{A}}$, $bhoT_{\mathcal{A}}$.

Remark: $inGr_{\mathcal{A}}$ is the English suffix "-ing", and used with English words.

$it_{\mathcal{A}}$: $\{y,\ u,\ t,\ smUH,\ rUp,\ r,\ o,\ n,\ mu/kt,\ mte,\ krN,\ k,\ it,\ e,\ ao,\ ai,\ IyA,\ I,\ HIn,\ A\}_{\mathcal{A}}$

Example bases: $p\_rbhAw_{\mathcal{A}}$, $niym_{\mathcal{A}}, An/nd_{\mathcal{A}}$.

Remark: $it_{\mathcal{A}}$ is used with nouns that can be converted to adjectives.

$kA_{\mathcal{A}}$: $\{zuri,\ t,\ r,\ o,\ n,\ l,\ kA,\ i,\ e,\ dAn,\ ch,\ b,\ N,\ HIn,\ E,\ A,\ /sth\}_{\mathcal{A}}$

Example bases: $tAr_{\mathcal{A}}$, $tuli_{\mathcal{A}}, bhUmi_{\mathcal{A}}$.

Remark: $kA_{\mathcal{A}}$ is an invalid suffix.

$k\_rme_{\mathcal{A}}$: $\{y,\ t,\ smUH,\ r,\ pt\_r,\ o,\ n,\ mte,\ m\hat{r}me,\ mUlk,\ lE,\ l,\ k\_rme,\ kAr,\ k,\ joge,\ e,\ bor,\ ao,\ ai,\ To,\ He\}_{\mathcal{A}}$

Example bases: $si/ddhA/nt_{\mathcal{A}}$, $ni\hat{r}desh_{\mathcal{A}}$.

Remark: $k\_rme_{\mathcal{A}}$ is used with words that can be used as to denote some plan.

$ke_{\mathcal{A}}$: $\{yo,\ t,\ sH,\ r,\ pti,\ lE,\ khn,\ ke,\ e,\ ao,\ ai,\ He\}_{\mathcal{A}}$

Example bases: $kAilE_{\mathcal{A}}$, $bidhAnsbhA_{\mathcal{A}}$, $pribeshTo_{\mathcal{A}}$.

Remark: In most cases $ke_{\mathcal{A}}$ is in fact the composite suffix $k + e_{\mathcal{A}}$. Hence it is not a suitable pivot suffix.

$keiTA_{\mathcal{A}}$: $\{zuri,\ zorA,\ ye,\ y,\ t,\ smUH,\ rUp,\ r,\ pt\_r,\ o,\ m\hat{r}me,\ m,\ lE,\ l,$
$keiTA,\ k,\ joge,\ i,\ e,\ dAn,\ bor,\ bAsI,\ ao,\ ai,\ To,\ N,\ Iy,\ I,\ He,\ HIn,\ A\}_{\mathcal{A}}$
Example bases: $bchr_{\mathcal{A}}$, $din_{\mathcal{A}}$.
Remark: $keiTA_{\mathcal{A}}$ is used with countable nouns with which the determiner $To_{\mathcal{A}}$ can be used.

$ne_{\mathcal{A}}$: $\{tA,\ re,\ r,\ o,\ ne,\ lE,\ i,\ ai,\ Hi,\ A\}_{\mathcal{A}}$
Example bases: $nkrib_{\mathcal{A}}$, $bi/shwAsjogJ_{\mathcal{A}}$, $zAne_{\mathcal{A}}$, $pArileH * eten_{\mathcal{A}}$.
Remark: $ne_{\mathcal{A}}$ is used to indicate inquisition. It can be used with different words. Hence it is not a suitable pivot suffix.

$owA_{\mathcal{A}}$: $\{wAn,\ w,\ u,\ tA,\ t,\ smUH,\ sbhA,\ rAz,\ r,\ p\hat{r}j/nt,\ owA,\ o,\ ni,\ n,\ mukhI,$
$mAn,\ m/nt\_rI,\ m,\ l,\ khn,\ k,\ ichil,\ i,\ e/shwr,\ e,\ din,\ dAn,\ ch,\ bor,\ birodhI,$
$bAsI,\ ai,\ To,\ J,\ I,\ He,\ E,\ A\}_{\mathcal{A}}$
Example bases: $zn_{\mathcal{A}}$, $khel_{\mathcal{A}}$.
Remark: $owA_{\mathcal{A}}$ is used with verbs that end with a consonant, in the simple present tense form.

$p\_rsAd_{\mathcal{A}}$: $\{zJoti,\ y,\ we,\ u,\ tA,\ t,\ rAz,\ r,\ p\_rsAd,\ nAth,\ k,\ i,\ e/shwr,\ e,\ bAd,\ ai,$
$To,\ I,\ A\}_{\mathcal{A}}$
Example bases: $dew_{\mathcal{A}}$, $stJ_{\mathcal{A}}$.
Remark: $p\_rsAd_{\mathcal{A}}$ is in fact a compound part, used with nouns that denote some revered entity.

$s/mp/nn_{\mathcal{A}}$: $\{wAn,\ t,\ smUH,\ sH,\ s/mp/nn,\ r,\ pU\hat{r}N,\ o,\ mAn,\ l,\ khini,\ k,\ ju/kt,$
$g,\ e,\ bor,\ biHIn,\ ao,\ ai,\ I,\ HIn,\ A\}_{\mathcal{A}}$
Example bases: $guN_{\mathcal{A}}$, $p\_rtibhA_{\mathcal{A}}$.
Remark: $s/mp/nn_{\mathcal{A}}$ is used with nouns that denote some quality.

$yA_{\mathcal{A}}$: $\{yA,\ skl,\ r,\ n,\ ch\}_{\mathcal{A}}$
Example bases: $zurI_{\mathcal{A}}$, $ngrI_{\mathcal{A}}$.
Remark: $yA_{\mathcal{A}}$ is an invalid suffix.

Though the experimental results show reasonably good classification of the base words, several issues become clear too. Some pivot suffixes are invalid, and

some others are not suitable pivot suffixes. Again, in most cases there are no base words to which all the suffixes in a given simple co-occurrence sets of a pivot suffix can be applied. Some of the drawbacks are due to noise in the input evidence prepared by an unsupervised method, while others are due to ambiguity of words.

6.6.2 Support of suffix co-occurrence

A common manifestation of noise in the suffixation evidence is in the form a suffix incorrectly associated to some base. This results in suffix characteristics with spurious elements, which in turn, causes spurious elements in the simple co-occurrence sets. An approach that we adopt to reduce such effects is to record the co-occurrence of two suffixes only if their co-occurrence is seen in at least a threshold number of bases. We refer to this threshold as the *minimum support* of the co-occurrence. The idea is that while a valid co-occurrences will be seen in many bases, co-occurrence of a suffix-pair due to noise will usually be seen only in very few bases. Hence insisting on a minimum support for co-occurrence can filter out most of the invalid co-occurrence. Like other support-based criteria, this too runs the risk of disqualifying valid cases. Hence, the minimum co-occurrence support value selected should not be too high. In our experiments reported below, we have tried using minimum co-occurrence support value of 2 besides the default value 1.

6.6.3 Theoretical weaknesses of the model

Some of the problems of the pivot suffix based classification arise because pivot suffixes do not cover all the situations that the set-theoretic model can present. Consider the situation for a hypothetical language L' as shown in Figure 6.2. Compared to language L (see Figure 6.1), in L', c does not exist, there is another suffix j which has b, e, f and g as co-occurring suffixes, and there is a category of words that takes only suffix f. There is no single pivot suffix for the categories corresponding to ellipses 3 and 6, and our algorithm fails to detect these categories. In fact we get the pivot suffixes a, b and h and their simple

co-occurrence sets { *a, d, e, f* }, { *b, f, e, j* } and { *h, g, i* }.

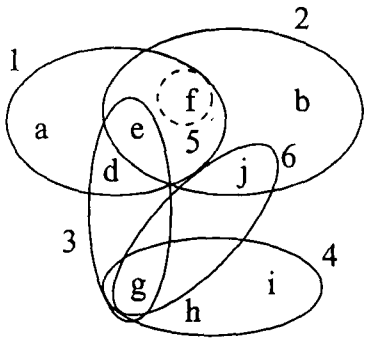


Figure 6.2: Suffixes and linguistic categories of words for language L'

Some other such complex situations are depicted in Figure 6.3.

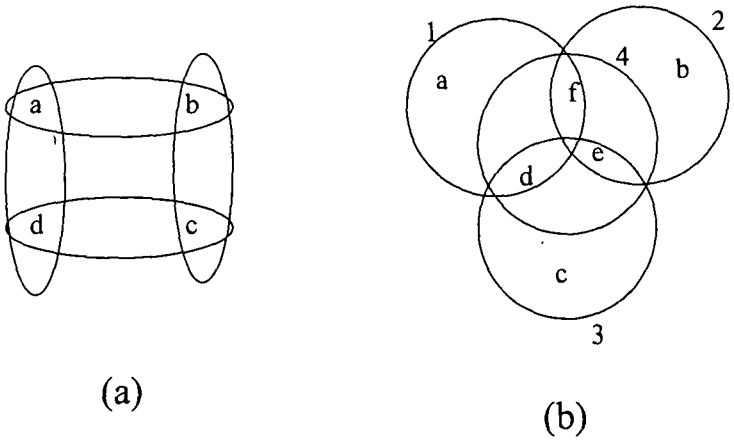


Figure 6.3: Complex co-occurrences

The important point to be observed is that the simple co-occurrence set of an individual suffix cannot be considered as the set of suffixes for a category of words. In the following section, we describe a more general consolidation of suffix co-occurrence evidence for identifying word categories.

6.7 Complete co-occurrence sets

In the procedure described in section 6.5, a suffix characteristic, C, is extended by taking an intersection of the simple co-occurrence sets of all the suffixes in C. Such extended co-occurrence sets might contain suffixes that do not co-occur with each other. This is illustrated in example 6.1

Example 6.1. Suppose the input evidence provides us the following suffix characteristics for the bases $a_s, b_s, c_s, d_s \& e_s$:

$$\begin{array}{lllll} a_s & : \{ & s_1, & s_2 \} & \\ b_s & : \{ & s_1, & & s_3 \} \\ c_s & : \{ & & s_2, & & s_4 \} \\ d_s & : \{ & & & s_3, & s_4 \} \\ e_s & : \{ & & s_2, & s_3 \}. \end{array}$$

Then the extended characteristic for e_s will be

$$\{s_1, s_2, s_3, s_4\}$$

though there is no evidence of co-occurrence of s_1 and s_4.

The pivot suffix based classification approach too has a similar problem. The simple co-occurrence lists (or sets) are computed by considering for each suffix the set of suffixes that have co-occurred with it. Simple co-occurrence sets might contain non-co-occurring suffixes, as illustrated in example 6.2.

Example 6.2. Suppose the input evidence has the words $as_1, as_2, bs_1, bs_3, etc.$ Then the simple co-occurrence set of s_1 will be

$$C^s(s_1) = \{s_1, s_2, s_3\}.$$

If s_1 is eventually identified as a pivot suffix, then the suffix-set $\{s_1, s_2, s_3\}$ will be taken as the set of suffixes for a category of words, whereas there is probably no evidence of co-occurrence of s_2 and s_3. So, it may not be useful to consider the simple co-occurrence list (of s_1, in this case) as the set of suffixes that a particular category of words takes.

The set of suffixes for a word-category should be a co-occurrence set such that every suffix in it has co-occurred with every other suffix in it. Let us call such a co-occurrence set a *complete co-occurrence* set or list. We symbolically denote a

complete co-occurrence set by an upper-case Roman letter with the superscript c. Each set containing a single suffix is complete co-occurrence set, but that is trivial. We are interested in co-occurrence sets that contain all the suffixes that the words of a given category take. More precisely, we are interested in *maximal complete co-occurrence* (MCC) sets of suffixes, *i.e.*, complete co-occurrence sets to which no suffix can be added without violating the complete co-occurrence property. This implies that an MCC set cannot be a subset of another distinct MCC set. Each MCC set can be considered to represent a distinct word-category. A word can be assigned to a category if its suffix characteristic is a subset of the MCC set corresponding to that category. Since distinct MCC sets can overlap, words occurring with inadequate number of suffixes may be assigned to more than one category We need to find sufficient MCC sets so that every word can be assigned to some category.

One possible approach to obtain a collection of MCC sets for the entire set of words in the training corpus is to consider each of the distinct suffix-characteristics of the words in non-increasing order of their sizes, and compute an MCC set that is a superset of that characteristic. We, however, need not consider a suffix-characteristic that is a subset of an already computed MCC set. To obtain an MCC set that is a superset of a suffix characteristic, C, we first prepare the list, L, of additional suffixes that will be there in the extended characteristic of C (see section 6.5). Then, we augment C by successively adding from L one suffix so that the complete co-occurrence property is not violated in the augmented set C. When all suffixes in L are considered, the augmented set C obtained is an MCC set that we require.

The collection of MCC sets obtained by the above method is adequate for the given training corpus, but *arbitrary*. More MCC sets can be possible. In the following section we present a general approach to compute *all* the maximal complete co-occurrence sets from the available co-occurrence evidence.

6.8 Computing all maximal complete co-occurrence sets

Let the suffixes under consideration be arranged in an order so that we can uniquely refer to each by its position in the order, *eg.*, the i^{th} suffix. Suppose the set of suffixes is

$$S = \{s_1, s_2, ..., s_n\}$$

so that the i^{th} suffix is s_i. Let us define a *k-maximal complete co-occurrence set* (k-mcc set), S^k, as a maximal subset of S such that all suffixes in the range $[s_1, s_k]$ present in it co-occur with all the suffixes in it. In particular, we define $S^0 = S$. Our target is to obtain all the S^n sets. The main idea of our method is to start with S^0 and proceed successively through $S^1, S^2, ..., S^n$ sets. While S^0 is unique (it is S), there can be multiple S^i for each $i = 1...n$. That is, in general, S^k is not unique. We maintain a list L of *k-mcc* sets. (In practice, each element in the list is stored as the pair (S^k, k), since the value of k is not obvious in the set S^k.) Initially L contains S^0 which is equal to S. One by one we take out a *k-mcc* set from L with $k = i$. If $i = n$, the *k-mcc* is a maximal complete co-occurrence set and it is produced as output. Else, from the *k-mcc* we compute *k-mcc* sets with $k = (i + 1)$ and include them in L. This procedure, referred to as **procedure all_mcc**, is described below:

Procedure all_mcc

> Let L be a set of pairs (K, i) where K is a *k-mcc* set with $k = i$. That is, K is a set S^i.
>
> Let $\overline{C^s}(\sigma)$ denote the set of suffixes in S that do not co-occur with suffix σ. Recall that $C^s(\sigma)$ is the simple co-occurrence set of suffix σ (section 6.4). That is
>
> $$\overline{C^s}(\sigma) = S - C^s(\sigma).$$

1. Initialise $L = \{(S, 0)\}$.
2. If L is empty, goto step 12; endif.
3. Pick a pair (K, i) from L, and remove it from L.
4. If $i = n$ then output K; goto step 2; endif.

5. If K does not contain the suffix s_{i+1} then add $(K, i+1)$ to L; goto step 2; endif.
6. Compute $Q = K \cap \overline{C^s}(s_{i+1})$, the set of suffixes in K that do not co-occur with s_{i+1}.
7. If $Q = \phi$ then add $(K \cup \{s_{i+1}\}, i+1)$ to L; goto step 2; endif.
8. Compute sets A and B as:

$$A = K - \overline{C^s}(s_{i+1}), \tag{6.1}$$

$$\text{and} \quad B = K - \{s_{i+1}\}. \tag{6.2}$$

9. If

$$\forall j \leq (i+1), \quad s_j \notin A \Rightarrow (A \cap \overline{C^s}(s_j)) \neq \phi \tag{6.3}$$

then add $(A, i+1)$ to L; endif.

10. If

$$\forall j \leq (i+1), \quad s_j \notin B \Rightarrow (B \cap \overline{C^s}(s_j)) \neq \phi \tag{6.4}$$

then add $(B, i+1)$ to L; endif.

11. Goto step 2.
12. End.

Explanation: Procedure *all_mcc* is an iterative procedure where steps 2 to 11 is one pass of the iteration. In one pass of the iteration, from a S^i set K with $i < n$, in L we obtain 0, 1 or 2 S^{i+1} sets and include them in L. If K does not contain the suffix s_{i+1}, then in step 5 K itself is the single S^{i+1} set obtained. Else, if suffix s_{i+1} co-occurs with all suffixes in K, then, $(K \cup \{s_{i+1}\}, i+1)$ is the single S^{i+1} set obtained (step 7). Else we consider two possible subsets of K in step 8– set A where the suffixes that do not co-occur with suffix s_{i+1} are excluded, and set B where suffix s_{i+1} is excluded. Clearly, all suffixes up to $i+1$ present in sets A and B co-occur with the rest of the suffixes present in them, respectively. Now, set A is a *k-mcc* set with $k = (i+1)$, if A contains every suffix $s_j, j \leq (i+1)$, such that s_j co-occurs with all suffixes in A. In other words, set A is a *k-mcc* set if for all $j \leq (i+1)$, s_j is not in A only if A contains some suffix that does not co-occur with s_j. For set A this is tested in step 9. Similarly, for B the analogous condition is tested in step 10. If sets A and B are found to be *k-mcc* sets respectively, they are included in L. Otherwise, they are discarded since

they are not maximal and by dropping more suffixes from them in subsequent iterations, no maximal complete co-occurrence sets can be obtained.

To show that the procedure *all_mcc* produces all the maximal complete co-occurrence sets of S, we need to show that-

1. *k-mcc* sets with $k = n$ are maximal complete co-occurrence sets,
2. the procedure correctly produces (k+1)-*mcc* sets from a *k-mcc* set.
3. the procedure produces all possible distinct *k-mcc* sets with $k = n$.

The first point above is satisfied since the definition of *k-mcc* sets (see page 136) implies that for $k = n$, a *k-mcc* set is a maximal complete co-occurrence set described in section 6.7. Similarly, the second point is satisfied from the arguments associated with the steps of the procedure. To establish the third point, we show that if X is a maximal complete co-occurrence set, procedure *all_mcc* will produce it. For this we trace the steps that the procedure will go through leading to X. Initially, we have S^0 containing all the suffixes, from which we can have one S^1 with s_1 and another without s_1. To trace the formation of X, we choose first or the second S^1 depending on whether X contains s_1 or not. If X does not contain s_1, we consider the second S^1 that does not contain s_1, but contains all the other suffixes. Else, if X contains s_1, it cannot have any suffix that does not co-occur with s_1, and we consider the first S^1 that contains s_1 and all other suffixes that co-occur with s_1. Thus $S^1 \supseteq X$. If S^1 we are considering does not contain s_2, X will also not have s_2, and according to the procedure we consider S^1 as S^2. Otherwise, if the S^1, contains s_2, in the next step there are two possibilities for S^2– with s_2 or without s_2. We select S^2 according to whether X contains s_2 or not. As usual, if S^2 has s_2, it will not have any suffix that does not co-occur with s_2, and X also cannot have any suffix that does not co-occur with s_2. In general, at any stage i, $S^i \supseteq X$ so that in the subsequent step we can have S^{i+1} according to whether X contains s_{i+1} or not. Proceeding in this way we get the sequence of S^i's so that finally S^n is same as X. The two choices of S^{i+1} at any stage correspond to the sets A and B in equations 6.1 and 6.2. If at any stage one of A and B cannot be retained according to conditions 6.3 or 6.4, it means at least one more suffix s_j, $j \leq i$, that co-occurs with all the other suffixes in that A or B, can be included in it. To trace the formation of X if it

is required to select such an A or B, it should be possible to include s_j in X too, since $X \subseteq S^1$. However, since X is a maximal complete co-occurrence set, that would be a contradiction. Hence, it will not be necessary to select A or B for which the conditions 6.3 or 6.4 respectively, are not satisfied.

Complexity of procedure *all_mcc*

Finding the all the maximal complete co-occurrence sets is equivalent to the *clique enumeration problem*, which is NP-hard. Procedure *all_mcc* is equivalent to a binary-tree building procedure where nodes at depth i correspond to *k-mcc* sets with $k = i$. The root corresponds to the *0-mcc* set, and the leaf nodes at depth n correspond to the *k-mcc* sets with $k = n$, *i.e.*, the maximal complete co-occurrence sets. First the simple co-occurrence sets for each of the n suffixes are computed by considering decompositions of m words. This requires effort of order $O(m * n)$ (section 6.6). Then starting from the root, the tree is built in a top-down fashion, *visiting* each node. The maximum possible number of nodes in the tree is $2^n + 1$. At each node, using the already computed simple co-occurrence sets,

computation of Q is of order	: $O(n)$
computation of A by equation 6.1 is of order	: $O(n)$
computation of B by equation 6.2 is of order	: $O(n)$
computation of condition 6.3 is of order	: $O(n^2)$
computation of condition 6.4 is of order	: $O(n^2)$

Hence, the total computation at each node is-

$$3 * O(n) + 2 * O(n^2).$$

The asymptotic estimation of the above is $O(n^2)$. For the entire procedure *all_mcc* the computation required is the sum of the computation of the simple co-occurrence sets and then the computation of the tree. This is equal to

$$O(m * n) + O(2^n * n^2).$$

Assuming that the 2^n is larger than m, the asymptotic complexity of procedure *all_mcc* is $O(2^n * n^2)$.

Though the above estimated complexity of procedure *all_mcc* is very high, in practice the computation required is much less. First, in the computation of the simple co-occurrence sets the effort required corresponding to a suffix characteristic $\imath$ with $p_\imath$ suffixes is actually of the order $O(p_\imath^2)$ and not $O(n^2)$. Hence, the computation of the simple co-occurrence sets is of the order $\sum_{\imath=1}^{m} p_\imath^2$. Again, in the process of computing the *k-mcc* sets, that is equivalent to the construction of a binary tree, a complete binary tree is not computed. Branches of the tree are pruned in several conditions. While computing $S^{\imath+1}$ from $S^\imath$, if $s_{\imath+1}$ is not present in $S^\imath$, or when Q is empty, only one $S^{\imath+1}$ is obtained, *i.e.*, only one child is created for the node corresponding to $S^\imath$. Again, if conditions 6.3 or 6.4 is not satisfied, the node corresponding to A or B respectively, is dropped. The occurrence of these conditions depends on the overall co-occurrence pattern of the suffixes in a complex way.

Despite being a computationally expensive, procedure *all_mcc* can be useful since it is required only in the training phase of classification, which can be called an "off-line" exercise. In dealing with test input, it need not be performed.

Experimental results

We have carried out experiments for finding out the maximal complete co-occurrence sets over the input described in page 118. Since the input is a morphological lexicon built with an unsupervised method, it contains invalid suffixes as well as invalid decompositions too. The number of maximal complete co-occurrence sets obtained is 2167, when the minimum co-occurrence support value used is 1. Few of these sets, for instance, are

$\{t,\ smUH,\ sH,\ r,\ o,\ nı,\ lE,\ khn,\ k,\ ı,\ e/shwr,\ e,\ bor,\ bırodhI,\ bılAk,\ To,\ Iy,\ I,$
$E,\ A,\ \#,\ \}_A$

$\{y,\ t,\ r,\ mu/kt,\ kAr,\ k,\ ju/kt,\ ı,\ To,\ HIn,\ /g\_r/st,\ \}_A$

$\{u,\ t,\ r,\ prj/nt,\ o,\ l,\ k,\ ichil,\ ı,\ e,\ To,\ I,\ Ar,\ A,\ \#ı,\ \}_A$

$\{t,\ s\#khJk,\ r,\ o,\ k,\ ı,\ e,\ bor,\ bhAwe,\ I,\ Ar,\ A,\ \}_A.$

With minimum co-occurrence support value 2, the number of maximal complete co-occurrence sets is 351. When minimum co-occurrence support value

is greater than 1, then it is likely that some suffix characteristics are not covered by any MCC sets. These are characteristics that contain weak co-occurring suffix-pairs.

To get an idea of the computational effort spent in procedure *all_mcc* we observe the following statistics of the tree construction when minimum co-occurrence support value is 1:

(a) No. of suffixes considered	:	122
(b) No. of MCC sets identified	:	2167
(c) No. of non-MCC nodes visited in tree	:	38854
(d) No. of times suffix s_{i+1} is absent in S^i	:	190291
(e) No. of times suffix s_{i+1} co-occurs with all of S^i	:	49495
(f) No. of times set A is dropped	:	7863
(g) No. of times set B is dropped	:	26658
(h) No. of times both sets A&B are dropped	:	5400.

Total number of nodes in the tree is the sum of the number of non-MCC nodes and the nodes corresponding to the MCC-set, *i.e.*, $2167 + 38854 = 41021$. This is quite an acceptable figure for 122 suffixes, considering that the procedure *all_mcc* tackles an NP-hard problem. Statements (d) and (e) gives the number of times S^i is taken as S^{i+1} as the only child of a node. (f) and (g) too, indicates number of times only one child is obtained for a node. Statement (h) indicates number of times when a *k-mcc* set is totally abandoned since it cannot lead to any MCC set. The count in statement (d) is significant because, in our implementation, when suffix s_{i+1} is absent in S^i, we simply reuse the node corresponding to S^i for S^{i+1} instead of creating a new node. This saves the creation of many new nodes in the process.

The number of MCC sets obtained is very high, particularly when the minimum co-occurrence support value is 1. This number is even higher than the number of distinct suffix characteristics, 1987, mentioned in section 6.3.1. Obviously, there are more MCC sets than is required for the input characteristics. In section 6.8.1 we discuss an approach for selecting essential MCC sets.

6.8.1 Essential maximal complete co-occurrence sets

Each MCC set has a unique composition of suffixes, which we consider as the distinctive set of suffixes of a category of words. No MCC set is a subset of another MCC set. MCC sets are formed by consolidating co-occurrence evidence from suffix characteristics of *several base words*, in such a way that each suffix characteristic is covered by at least one MCC set. It is possible that no *single* characteristic exactly matches a particular MCC set. If a base word has occurred with adequate number of suffixes, its suffix characteristic is exclusively covered by only one MCC set, and it can be classified into an unique word class. Otherwise, its suffix-characteristic is a subset of multiple MCC sets.

Procedure *all_mcc* in section 6.8, identifies more MCC sets than is required to cover all the input suffix characteristics. Many of these probably represent only theoretical word classes. We are interested in selecting the minimum number of MCC sets that can cover all the input characteristics. For this we carry out the following steps:

1. Retain the MCC sets that exclusively cover some suffix characteristics. We refer to these MCC sets as $M1$ MCC sets. For minimum co-occurrence support 2, the no of $M1$ MCC sets is 87.

2. We refer to the suffix characteristics that are not covered by $M1$ MCC sets as $S2$ characteristics. For the $S2$ characteristics we find MCC sets that cover them. We refer to these as $M2$ MCC sets. Since $S2$ sets are not exclusively covered by any MCC set, hence there shall be more than one MCC sets that cover each $S2$ set. So, for identifying $M2$ MCC sets there can be the following possibilities–

   (a) Identify all MCC sets that cover an $S2$ characteristic set. The advantage of this criteria is that for the $S2$ sets all possibilities without violating the input co-occurrence evidence are open. The drawback is that the number of MCC sets selected is large and there is much redundancy. In our experiment with minimum co-occurrence support value 2, the number of $M2$ MCC sets identified like this is 219.

(b) For an $S2$ characteristic set that is not covered by any of the already identified $M2$ set, select the largest MCC set that covers it. The idea is that a large MCC set is likely to cover other $S2$ sets too, so that the total no of $M2$ sets identified is less. In our experiment with minimum co-occurrence support value 2, the number of $M2$ MCC sets identified like this is 82.

(c) Let us refer to the $S2$ characteristics that are not subsets of other $S2$ characteristics, as $S3$ sets. For each non-$M1$ MCC set find the number of $S3$ sets that it covers. Then iteratively select as $M2$ the MCC set that covers the largest number of remaining $S3$ sets, till all the $S3$ sets are covered. This is a greedy approach to select the minimum number of $M2$ sets. It restricts the classification possibilities for the $S2$ sets, but selects the minimum number of MCC sets, by retaining only the essential MCC sets. In our experiment with minimum co-occurrence support value 2, the number of $M2$ MCC sets identified is 35.

In another experiment of identifying the essential MCC sets, we take as input the suffixation evidence in the lexicon obtained from corpus A of over 1,16,000 words (see page 81). We consider 102 suffixes with occurrence frequency greater than or equal to 7. With minimum co-occurrence support 2, the results obtained can be summarised as–

Total MCC sets obtained	: 208
Essential MCC sets:	
$M1$ sets	: 83
$M2$ sets by approach 2a	: 102
$M2$ sets by approach 2b	: 35
$M2$ sets by approach 2c	: 16.

6.9 Minimal signatures of word categories

Once the underlying words categories in a language are identified in terms of the set of suffixes that can be applied to words of that category, it is possible to classify words into these categories by considering the suffixes that are seen

with that word. However, since the sets of suffixes corresponding to different word categories may be overlapping, and each base word in a given input generally occur with only a subset of all the possible suffixes for its category, the classification may be in-exact. Let us refer to a set of suffixes that can be uniquely associated with a word category as a *signature* of the category. A word can be exclusively classified into a category if it has occurred with the signature of the category.

The complete set of suffixes that can occur with words of a given category, is a signature of the category. Apart from that there can be subsets of suffixes that are signatures of the word categories. For example, in the pivot suffix model of word classification (section 6.6), each pivot suffix and co-pivot suffix is a signature of the word category it represents. Suppose, $C_1, C_2, ..., C_n$ are the complete sets of suffixes associated with n different categories of words respectively. A set of suffixes S_i is a signature of the class i if $S_i \subseteq C_i$ and $S_i \not\subseteq C_j$, $j \neq i$. In general, there may not be any single-suffix signature of a word category. Instead, it is possible to identify *minimal* sets of suffixes that are signatures. We refer to these as *minimal signatures* of word categories. A signature S_i of category i is minimal if no proper subset of S_i is a signature of category i. There may be more than one minimal signatures for each word category. Finding all the minimal signatures of word category is a computationally expensive exercise. In section C.4 we describe a general procedure for finding all the minimal signatures of word categories.

It must be pointed out that since the number of minimal signatures can be more than the number of word categories, they do not enhance the effectiveness or efficiency of word classification. Suppose the number of word categories is N and the number of minimal signatures is R, so that $N \leq R$. To determine if a word can be exclusively put in one of the categories, we have to see if its suffix characteristic is a subset of *exactly one* of the N sets of suffixes corresponding to the N categories. The effort required for this is of order $O(N)$. Alternatively, if we consider the minimal signatures for the purpose, then we have to see if any of the R minimal signatures is a subset of the suffix characteristic of the word. The effort required will be of the order $O(R)$ (for successful cases the effort required will be on an average of order $O(R/2)$).

6.10 Shortcomings of suffix based word classification

Unsupervised methods for identifying word categories from suffixation evidence depends heavily on suffix characteristics of individual words and co-occurrences of suffixes. Hence, base words that are inherently ambiguous about their category make the classification task difficult. For example, the word $khel_{\text{অ}}$ (খেল) is ambiguous since it can be used as a noun to mean a "game" and as a verb to mean "play". In an input text, $khel_{\text{অ}}$ may occur with suffixes for nouns as well as suffixes for verbs. If that happens in the training corpus, it will appear that $khn_{\text{অ}}$, which is a suffix for nouns, co-occurs with $ichil_{\text{অ}}$, which is a suffix for verbs. That is, the suffix characteristic of $khel_{\text{অ}}$ may imply suffix co-occurrences that are actually *exceptions*, which should not be generalised. Again, there are certain suffixes in Assamese, particularly the determiners, whose applicability depends on some very subtle criteria. This leads to formation of too many word categories. Such fine categories of words reflect the morphological behaviour well, but it may make a subsequent exercise such as syntax modelling, difficult. We feel that it may be useful to group some such word categories according to their syntactic behaviour. Thus there will be some kind of hierarchical classification of words.

Another shortcoming of the whole idea of word classification based on suffix evidence is that there can be some categories of words that do not take any suffixes. For example, in Assamese, the words $Aru_{\text{অ}}, bA_{\text{অ}}, ki/ntu_{\text{অ}}$, (আৰু, বা, কিন্তু, meaning *and, or, but*), *etc.*, generally do not take any suffix. Suffix based methods may at most identify all such words as belonging to a single category. In a highly inflectional language like Assamese, where most of the root words may undergo suffixation, this is not a very serious problem. (All the above three words really belong to a single category - conjunctions.) However, one should be careful in such decisions because a word may appear without suffix only in the corpus being considered. A simple criteria can be - if the word has occurred a large number of times in the corpus, but always occurred without suffix, probably it never takes any suffix.

6.11 Summary

We have represented the problem of affix-based word classification in terms of sets and discussed some possible approaches for unsupervised identification of linguistic categories of words in a language using information of association of suffixes with different words in a corpus. Then we have described how, the words can be subsequently classified into the identified categories using the same input morphological evidence.

Chapter 7

Conclusions and Future Work

Morphological analysis is a very significant step of NLP for highly inflectional languages such as Assamese. Morphology and syntax are two complementary parts of the structural framework of natural language expression. And it is according to the structure of an expression that the overall primary meaning of the expression is composed from the implicit meanings of the individual elements of the expression. Because of the structural nature of morphology, simple computational methods can serve as the initial steps for acquisition of morphology of a language and morphological analysis. More efforts beyond the simple methods are required to tackle different *language specific* and *script specific* issues.

We have been largely successful in computational acquisition of the morphology of Assamese. We believe ours is the first such achievement, and hence pioneering. Our work is particularly significant because morphology is the dominant structural phenomenon in Assamese. The remaining structural analysis of Assamese texts can greatly benefit from the morphological analysis. One of the products of our work is a morphological lexicon for Assamese, which can be used in different end-user applications.

We have identified and tackled several issues inherent in using a raw text corpus for acquisition of morphology of a language. We have also tackled issues specific to the language, script and encoding schemes. The morphological knowledge acquired from the training corpus using the unsupervised approach,

is less than perfect, but we have developed methods that give fairly good results when this knowledge is subsequently applied for morphological analysis of input texts. In the training phase, from a corpus of about 300000 words derived from newspaper articles, where there are about 190 suffixes, we identified suffixes with a precision of 67% and a recall of about 72%. When this knowledge along with the lexicon created is used for morphological analysis of unseen newspaper articles, we achieve a precision and recall of over 85%. For texts from other domains, the precision of analysis obtained is over 85% and the recall is around 80%. Our method is more effective than other proposed methods that we have studied (*eg.*, [16, 19]). *This accomplishment can be considered an important milestone of NLP for Assamese.* The morphological analysis provided by our method can be directly used for further processing of Assamese texts, such as syntax and semantic processing.

The morphological structures of words provide clues regarding the categories of the words. Classification of words into such categories can be of great use in subsequent syntax analysis of the texts. However, care is required in drawing inferences from evidence provided by a corpus. We have defined novel set-theoretic approaches for classification of words based on their morphological behaviour, taking into account the complications that arise in these tasks. We have also presented sound theoretical explanations of the procedures involved. These give good results wherein words are put in more fine grained categories compared to usual word classes. However, some of the identified categories are unrealistic. From a corpus of about 300000 words we identified about 120 word categories, each of which is associated with a distinct set of suffixes.

We would like to remark that, since morphology evolves according to the spoken form of a language, for its unsupervised acquisition from a written corpus, it will be helpful if the script *clearly and unambiguously* reflects the phonological structure of the expressions. This depends on the orthography of the languages. We find that Assamese orthography is stronger than English. In English the pronunciations of words often cannot be accurately figured from their spellings alone. In Assamese it is not so. On the other hand, the Assamese script has lot of redundancy as far has its spoken form is concerned. Unsupervised acquisition of morphology would be easier if such redundancy can be removed.

A more pronunciation specific encoding may be useful in this regard. On that count Hindi, which uses the Devnagri script is better off. But Hindi is not as morphologically rich as is Assamese.

Future Work

Though significant, morphological analysis is only part of the larger problem of NLP. The work can be followed by the remaining NLP tasks. Also, there is scope for improvement in the tasks that we have accomplished. Varying degrees of supervision can be introduced to get more accurate results.

One immediate follow-up work can be implementation of a spelling-checker for the language we have considered, *i.e.*, Assamese. For a highly inflectional language a spelling-checker can be effective only if it has substantial morphological analysis capabilities. Also, the lexicon that we have produced is a useful resource for such purposes.

In word classification, there is scope for more realistic word categories, which are more suitable for subsequent tasks such as syntax analysis. Stronger measures to tackle the effect of noise and sparseness of evidence may be sought. One possibility is to consider suffix sequences information for classification, since compared to individual suffixes, suffix-sequences are generally less ambiguous and less noise-prone. It will also be interesting to try the pivot-suffix based method as well as the MCC based method on better quality suffixation evidence prepared by supervised methods.

In a task like NLP that is usually carried out in stages, results or insights gained in one stage can be useful in resolving issues of an earlier phase. In context of our work, we feel that the figured category attribute of a word can be used to improve the quality of decompositions related to that word, by ruling out candidate decompositions that are not valid for that category. Similarly, feedback from syntax analysis stage can provide hints for word decomposition as well as word classification. Experiments in this line can be taken up as an immediate extension of our work.

As mentioned in the previous section, writing systems that reflects the

phonology of words more realistically can improve unsupervised acquisition of morphology. Experiments with more phonetically driven encoding schemes can be carried out. Also, since different encoding schemes are in use for Assamese texts in computers, suitable transliteration software can be developed to inter-operate between these schemes. It will enhance the benefits obtained from work such as ours, making them more effective.

Appendix A

The Assamese Alphabet

A.1 The basic alphabet

The basic Assamese alphabet is traditionally presented in a tabular format as shown in Table A.1

	1	2	3	4	5
1	অ	আ	ই	ঈ	
2	উ	ঊ	ঋ		
3	এ	ঐ	ও	ঔ	
4	ক	খ	গ	ঘ	ঙ
5	চ	ছ	জ	ঝ	ঞ
6	ট	ঠ	ড	ঢ	ণ
7	ত	থ	দ	ধ	ন
8	প	ফ	ব	ভ	ম
9	য	ৰ	ল	ৱ	
10	শ	ষ	স	হ	
11	ক্ষ	ড়	ঢ়	য়	
12	ৎ	ং	ঃ	ঁ	

Table A 1: The basic Assamese alphabet

Rows 1-3 of the alphabet in Table A.1 are the vowels, and the rows 4-11

are consonants. The four symbols in row 12 are partial consonants. They can occur only associated with other letters. Thus their effect is not evident from their names (see pronunciation chart below). In addition, there are three more *consonant operators* - *ra-kAr* (্র, *eg.*, প্রবল, read as *prabal*), *ref* (র্, *eg.*, কর্ম, read as *karma*) and *ja-kAr* (্য, *eg.*, ব্যয়, read as *byay*), which can be associated with the consonants in rows 4-11. Their effect is to modify the pronunciation of the associated consonant. The Roman transcription used in this document for these operators are _r, r̂ and J, respectively.

Each of the vowels except the first vowel 'a' (অ), has a corresponding *operator* symbol as shown in Table A.2. In writing a word these operators may be associated with the consonants in rows 4-11. Without a vowel operator, the pronunciation of a consonant is either supported by the default (inherent) vowel 'a' (অ), or by the preceding vowel (possibly associated with a consonant). When a vowel operator is associated with a consonant the consonant is pronounced followed by that vowel.

	1	2	3	4
1	ক	কা	কি	কী
2	কু	কূ	কৃ	
3	কে	কৈ	কো	কৌ

(The consonant ক is used as an example)

Table A.2: The vowel operators

The approximate pronunciation of the letters are given below:

Sl.No.	Assamese letter	Roman transcription used in this document	read-as (approx.)	example
1.	অ	a	*o*	the *a* in *tall*
2.	আ	A	*aa*	the *a* in *part*
3.	ই	i	*hraswa-e*	the *i* in *bit*

4.	ঈ	I	*dirgha-e*	the *ee* in *feet*
5.	উ	u	*hraswa-oo*	the *u* in *pull*
6.	ঊ	U	*dirgha-oo*	the *oo* in *school*
7.	ঋ	Rh	*ri*	the *ri* in *Krishna*
8.	এ	e	*a*	the *a* in *pack*
9.	ঐ	E	*oi*	the *ai* in *Jain*
10.	ও	o	*o*	the *oa* in *coat*
11.	ঔ	O	*ou*	the *ow* in *rowed*
12.	ক	k	*ka*	the *ca* in *call*
13	খ	kh	*kha*	the *kha* in *Jharkhand*
14.	গ	g	*ga*	the *ga* in *gall*
15.	ঘ	gh	*gha*	the *gh* in *ghost*
16.	ঙ	nG	*unga*	the ng^1 in *hanger*
17.	চ	c	*pratham-sa*	the s^1 in *gas*
18.	ছ	C	*dwitiya-sa*	(similar to *pratham* − *sa*)
19.	জ	z	*bargiya-za*	the z^1 in *Amazon*
20.	ঝ	jh	*jha*	the Jh^1 in *Jharkhand*
21.	ঞ	nY	*nya*	the *ian* in *fiance*
22.	ট	T	*murdhanya-ta*	the *to* in *top*
23.	ঠ	Th	*murdhanya-tha*	the th^1 in *thousand*
24.	ড	D	*murdhanya-da*	the *do* in *doctor*
25.	ঢ	Dh	*murdhanya-dha*	the Dh^1 in *Dhaka*
26.	ণ	N	*murdhanya-na*	the n^1 in *Ganesh*
27.	ত	t	*dantya-ta*	(similar to murdhanya-ta)
28.	থ	th	*dantya-tha*	(similar to murdhanya-tha)
29.	দ	d	*dantya-da*	(similar to murdhanya-da)
30.	ধ	dh	*dantya-dha*	(similar to murdhanya-dha)
31.	ন	n	*dantya-na*	(similar to murdhanya-na)
32.	প	p	*pa*	the *po* in *point*
33.	ফ	ph	*pha*	the ph^1 in *phone*
34.	ব	b	*ba*	the *ba* in *ball*
35.	ভ	bh	*bha*	the Bh^1 in *Bharat*

36.	ম	m	*ma*	the *ma* in *mall*
37.	য	j	*ja*	the *jo* in *jog*
38.	ৰ	r	*ra*	the *ro* in *rock*
39.	ল	l	*la*	the *lo* in *lost*
40.	ৱ	w	*wabba*	the *wo* in *world*
41.	শ	sh	*talabya-sa*	(roughly) the sh^1 in *posh*
42.	ষ	S	*murdhanya-sa*	(roughly) the sh^1 in *posh*
43.	স	s	*dantya-sa*	(roughly) the sh^1 in *posh*
44.	হ	H	*ha*	the *ha* in *hall*
45.	ক্ষ	X	*khya*	(absent in English)
46.	ড়	R	*dare-ra*	the r^1 in *Orıssa*
47	ঢ়	rh	*dhare-ra*	the rh^1 in *Chandigarh*
48.	য়	y	*ya*	the *you* in *young*
49.	ৎ	_t	*byanjan-ta*	the *t* in *Utpal*
50.	ং	#	*anuswar*	the *ng* in *king*
51.	ঃ	:	*bisarga*	the *h* in *eh*
52.	ঁ	*	*sandra-bindu*	the *n* in *Ranchi*
53.	্র	_r	*ra-kAr*	the *r* in *product*
54.	র্	r̂	*ref*	the *r* in *form*
55.	্য	J	*ja-kAr*	the *y* in *Myanmaar*

[1] the inherent vowel 'a' is to be added.

A.2 The numerals

০	১	২	৩	৪	৫	৬	৭	৮	৯
0	1	2	3	4	5	6	7	8	9

Appendix B

Suffixed Forms of Assamese Nouns and Verbs

List A

1	*l'rA* (ল'ৰা)	2	*l'rATo* (ল'ৰাটো)
3	*l'rAzn* (ল'ৰাজন)	4	*l'rAkn* (ল'ৰাকন)
5	*l'rAbor* (ল'ৰাবোৰ)	6	*l'rAbɪlAk* (ল'ৰাখিনি)
7	*l'rAkhɪnɪ* (ল'ৰাখিনি)	8	*l'rAmkhA* (ল'ৰামখা)
9	*l'rAskl* (ল'ৰাসকল)	10	*l'rAsmUH* (ল'ৰাসমূহ)
11	*l'rAgAl* (ল'ৰাগাল)	12	*l'rAzAk* (ল'ৰাজাক)
13	*l'rAkeɪTA* (ল'ৰাকেইটা)	14	*l'rAkeɪzn* (ল'ৰাকেইজন)
15	*l'rAH*t* (ল'ৰাহঁত)		

List B

1	*l'rAɪ* (ল'ৰাই)	2	*l'rAmkhAɪ* (ল'ৰামখাই)
3	*l'rAkeɪTAɪ* (ল'ৰাকেইটাই)		

List C

1	*l'rATowe* (ল'ৰাটোৱে)	2	*l'rAzne* (ল'ৰাজনে)
3	*l'rAkne* (ল'ৰাকনে)	4	*l'rAbore* (ল'ৰাবোৰে)
5	*l'rAbɪlAke* (ল'ৰাবিলাকে)	6	*l'rAkhɪnɪye* (ল'ৰাখিনিযে)
7	*l'rAskle* (ল'ৰাসকলে)	8	*l'rAsmUHe* (ল'ৰাসমূহে)
9	*l'rAzAke* (ল'ৰাজাকে)	10	*l'rAgAle* (ল'ৰাগালে)
11	*l'rAkeɪzne* (ল'ৰাকেইজনে)	12	*l'rAH*te* (ল'ৰাহঁতে)

To each form in List A, the following (composite) suffixes can be appended:

-k (-ক)	*-kno* (-কনো)	*-kei* (-কেই)	*-kHe* (-কহে)
-klE (-কলৈ)	*-klEno* (-কলৈনো)	*-klEHe* (-কলৈহে)	*-keiHe* (-কেইহে)
-keino (-কেইনো)	*-kto* (-কতো)	*-keito* (-কেইতো)	*-r* (-ৰ)
-rno (-ৰনো)	*-rei* (-ৰেই)	*-rHe* (-ৰহে)	*-reiHe* (-ৰেইহে)
-rto (-ৰতো)	*-rtono* (-ৰতোনো)	*-reito* (-ৰেইতো)	*-rtoHe* (-ৰতোহে)
-lE (-লৈ)	*-lEno* (-লৈনো)	*-lEhe* (-লৈহে)	*-lEto* (-লৈতো)

To each form in List B, the following (composite) suffixes can be appended:

-yei (-য়েই) *-yeicon* (-য়েইচোন) *-yeino* (-য়েইনো) *-yeiHe* (-য়েইহে)
-yeito (-য়েইতো)

To each form in List C, the following (composite) suffixes can be appended:

-i (-ই) *-icon* (-ইচোন) *-ino* (-ইনো) *-iHe* (-ইহে)
-ito (-ইতো)

To each form in Lists A, B and C, the following (composite) suffixes can be appended:

-no (-নো) *-He* (-হে) *-con* (-চোন)

(Some other possible forms are omitted).

Total: $A + B + C + (A * 24) + (B * 5) + (C * 5) + ((A + B + C) * 3)$
i.e., $15 + 3 + 12 + 360 + 15 + 60 + 90 = 555$ forms

Table B.1: Suffixed forms of the noun *l'rA* (meaning *boy*)

List A

1	*bH* (বহ)	2.	*bHo** (বহোঁ)
3	*bHiCo* (বহিছো)	4.	*bHilo* (বহিলো)
5	*bHiCilo* (বহিছিলো)	6.	*bHim* (বহিম)
7	*bHA* (বহা)	8.	*bHiCA* (বহিছা)
9	*bHilA* (বহিলা)	10.	*bHiCilA* (বহিছিলা)
11	*bHibA* (বহিবা)	12	*bHk* (বহক)
13	*bHiCe* (বহিছে)	14.	*bHile* (বহিলে)
15	*bHiCile* (বহিছিলে)	16.	*bHib* (বহিব)
17	*bHe* (বহে)	18.	*bHil* (বহিল)
19	*bHAo** (বহাওঁ)	20.	*bHAiCo* (বহাইছো)
21	*bHAlo* (বহালো)	22.	*bHAiCilo* (বহাইছিলো)
23	*bHAm* (বহাম)	24.	*bHowA* (বহোৱা)
25	*bHAiCA* (বহাইছা)	26.	*bHAlA* (বহালা)
27	*bHAiCilA* (বহাইছিলা)	28.	*bHAbA* (বহাবা)
29	*bHAi* (বহাই)	30.	*bHAiCe* (বহাইছে)
31	*bHAle* (বহালে)	32.	*bHAiCile* (বহাইছিলে)
33	*bHAb* (বহাব)	34.	*bHuwAo** (বহুৱাওঁ)
35	*bHuwAiCo* (বহুৱাইছো)	36.	*bHuwAlo* (বহুৱালো)
37	*bHuwAiCilo* (বহুৱাইছিলো)	38.	*bHuwAm* (বহুৱাম)
39	*bHuwA* (বহুৱা)	40.	*bHuwAiCA* (বহুৱাইছা)
41	*bHuwAlA* (বহুৱালা)	42.	*bHuwAiCilA* (বহুৱাইছিলা)
43	*bHuwAbA* (বহুৱাবা)	44.	*bHuowA* (বহুওৱা)
45	*bHuwAi* (বহুৱাই)	46.	*bHuwAiCe* (বহুৱাইছে)
47	*bHuwAle* (বহুৱালে)	48.	*bHuwAb* (বহুৱাব)

List B

1	*bHcon* (বহচোন)	2	*bHo*con* (বহোঁচোন)
3	*bHAcon* (বহাচোন)	4.	*bHkcon* (বহকচোন)

List C

1	*bHiblE* (বহিবলৈ)	2	*bHiblEHe* (বহিবলৈহে)
3	*bHiblEto* (বহিবলৈতো)	4.	*bHiblEo* (বহিবলৈও)
5	*bHiblEno* (বহিবলৈনো)	6.	*bHiblEono* (বহিবলৈওনো)

7.	*bHiblEne* (বহিবলৈনে)	8.	*bHo*te* (বহোঁতে)
9.	*bHAt* (বহাত)	10.	*bHo*teo* (বহোঁতেও)
11.	*bHAto* (বহাতো)	12.	*bHo*teHe* (বহোঁতেহে)
13.	*bHAtHe* (বহাতহে)	14.	*bHo*teoto* (বহোঁতেওতো)
15.	*bHo*teje* (বহোঁতেযে)	16.	*bHAtje* (বহাতযে)
17.	*bHo*teoje* (বহোঁতেওযে)	18.	*bHAtoje* (বহাতোযে)
19.	*bHo*teno* (বহোঁতেনো)	20.	*bHAtno* (বহাতনো)
21.	*bHo*teono* (বহোঁতেওনো)	22.	*bHAtono* (বহাতোনো)
23.	*bHimcon* (বহিমচোন)	24.	*bHibAcon* (বহিবাচোন)
25.	*bHibcon* (বহিবচোন)	26.	*bHo*con* (বহোঁচোন)
27.	*bHAcon* (বহোচোন)	28.	*bHkcon* (বহকচোন)

To each form in List A, the following (composite) suffixes can be appended:

-gE (-গৈ)	*-ne* (-নে)	*-He* (-হে)	*-Hi* (-হি)
-gEcon (-গৈচোন)	*-Hicon* (-হিচোন)	*-gEHe* (-গৈহে)	*-to* (-তো)
-je (-যে)			

To each form in List B, the following suffixes can be appended:

-gE (-গৈ) *-Hi* (-হি)

(Some other possible forms are omitted).

Total: $A + B + C + (A * 9) + (B * 2)$

i.e., $48 + 4 + 28 + 432 + 8 = 520$ forms

Table B.2: Suffixed forms of the verb *bH* (meaning *sit*)

Appendix C

Implementation Outlines

C.1 Initial decompositions

A simple way to identify decompositions for words in a list with other words in that list as bases, is to first sort the words alphabetically. This would ensure that if a word can be decomposed then the corresponding bases would occur in the *preceding neighbourhood* in the sorted list. More specifically, the process can be stated as-

Algorithm 1:

1. Let input text be T

2 Form a sorted list, L, of distinct words in T.

3. For each word w_i in L, identify another word $w_j, j < i$ in L such that w_i can be obtained by appending some (non-null) suffix s to w_j. This gives the decomposition

$$[w_i = w_j + s]_s.$$

If no w_j can be identified for a w_i, w_i is "undecomposed".

C.2 Unifying decompositions

Suppose we have the set of initial decompositions, D. Each decomposition is of the form

$$[w = b + x]_s,$$

where w_s is the word being decomposed, b_s is the base, and x_s is a suffix. For some words there can be more than one decompositions in D each with a differed base-suffix pair. To obtain a single decomposition for such words by combining the multiple decompositions, the following steps can be performed:

Algorithm 2:

1. Sort the decompositions in D on the *word* field so that multiple decompositions of the same word occur together.

2. Scan the decompositions to identify words that have multiple decompositions.

3. If for a word, w_s, there is a single decomposition, output that decomposition.

4. Else, suppose there are n decompositions of a word, w_s.

5 For each of the n decompositions of w_s, initialise a "cursor" that points to the first letter of the base.

6. Iteratively till the end of the decompositions is reached, if one or more of the cursors are over a partition point (a '+' mark) in the respective decompositions, output a partition point, and advance only those cursors. Else, output the letter under the cursor and advance all the cursors by one letter.

C.3 Finding suffix-sequences

Suppose we have the set of initial decompositions, D. Each decomposition is of the form

$$[w = b + x]_s,$$

where w_s is the word being decomposed, b_s is the base, and x_s is a suffix. From D suffix-sequences can be identified using the following steps-

Algorithm 3:

1. Unify the decompositions in D to obtain a list of decompositions L, that is sorted on the word field.
2. Initialize list of output decompositions, M as EMPTY.
3. From beginning of L, for each decomposition do step 4.
4. Suppose from L we read the decomposition

   $$[w = b + x_1]_s.$$

   If there is a decomposition of the word b_1 in M as

   $$[b = b_1 + x_2]_s,$$

   put in M the decomposition

   $$[w = b_1 + x_2 + x_1]_s.$$

   Both x_{1_s} and x_{2_s} are single suffixes or themselves suffix sequences.
5. At the end M has decompositions involving suffix sequences.

C.4 Compute minimal signatures of suffixes of word categories

Suppose S is the set of suffixes $\{s_1, s_2, ..., s_n\}$, and $C_1, C_2, ..., C_m$ are the sets of the suffixes associated with distinct word categories, obtained from the given

suffixation evidence. We assume that $C_i \not\subseteq C_j$ for distinct values of i and j between 1 and m. That is, each C_i is a signature. However, it may be possible to drop some elements from C_i such that the reduced set is still a signature. Our objective is to find for each C_i, minimal sets of suffixes that are its signatures.

To find the minimal signatures of one of the suffix sets C_i a general approach is to partition C_i into two – partition A and partition B. Initially partition A is empty and $B = C_i$. Through a recursive procedure, we move selected suffixes from B to A till A is a signature. Until A becomes a signature, A is a subset of more than one distinct C_j. We say that a suffix in B is *significant* if upon including it in A, A becomes the subset of fewer distinct C_j. The outline of the recursive procedure is given below as *procedure* $min\_sign(A, B)$:

Algorithm 4:

```
    procedure min_sign(A, B)
1.  begin
2.      while A is not a signature
3.          remove a suffix s from B
4.          if s is not significant
5.              proceed to next iteration
6.          else
7.              if (A ∪ B) is a signature     // B does not have s now
8.                  min_sign(A, B);
9.              endif
10.             include s in A
11.             proceed to next iteration
12.         endif
13.     endwhile
14.     if A is not superset of already declared minimal signature
15.         declare A as a minimal signature;
16.     endif
17. end
```

Procedure $min\_sign(A, B)$ is equivalent to a binary-tree building process in top-down fashion. A distinct pair (A, B) is associated to each node. At the root, A is empty and $B = C_i$. By the *while-loop* we traverse down from the root through the left-children to a leaf. In step 8, we create a new sub-tree through recursion, with its root as the right-child of the current node. In step 11, we proceed to the other child of the current node, to repeat the exercise. If there are p suffixes in C_i, the number of nodes in the tree can be up to $2^{p+1} - 1$. In practice, for our given problem, the number is far less.

To decide whether a set X is a signature we need to determine if X is a subset of exactly one of the sets $C_1, C_2, ..., C_m$. If X has p number of suffixes, the computation required for this is of the order $O(p*m)$. Similarly, to decide whether suffix s is significant, we have to count the number of sets $C_1, C_2, ..., C_m$ of which A is a subset, and repeat the same after including s in A (for testing this for another suffix and same A, we have to compute only the second count and reuse the first count). The computation required for this is of the order of $O(p * m)$. When the suffix sets associated with the word categories are the maximal complete co-occurrence (MCC) sets $C_1^c, C_2^c, ..., C_m^c$, these computations can be improved by using the certain properties of MCC sets. Suppose $C_1^s, C_2^s, ..., C_n^s$ are the simple co-occurrence sets of the suffixes $s_1, s_2, ..., s_n$, respectively. Then the relevant properties of MCC sets are:

1. The intersection of the simple co-occurrence sets of the suffixes in an MCC set C_i^c is the set C_i^c itself. That is,

$$C_i^c = \bigcap_{\forall s_j \in C_i^c} C^s(s_j).$$

   [Recall that $C^s(\sigma)$ is the simple co-occurrence set of suffix σ.]

2. If a set of suffixes S_i is a signature of a word category corresponding to C_i^c, then the intersection of the simple co-occurrence sets of the suffixes in S_i is the set C_i^c itself. That is,

$$C_i^c = \bigcap_{\forall s_j \in S_i} C^s(s_j).$$

3. If a subset S_i of C_i^c is not a signature of the word category corresponding to C_i^c, then the intersection of the simple co-occurrence sets of the suffixes in S_i is a superset of C_i^c. That is,

$$C_i^c \subset \bigcap_{\forall s_j \in S_i} C^s(s_j).$$

In actual implementation, for set A we maintain a set A^I, which is the intersection of the simple co-occurrence set of the suffixes in it. In particular, when A is empty initially, we initialise A^I as S. Every time a suffix s from B is included in A, A^I is updated to its intersection with the simple co-occurrence set of s. This requires a computation of order $O(n)$. To test whether A is a signature (the condition in the *while* statement), we test whether $A^I = C_i^c$. This too, requires a computation of order $O(n)$. To test whether suffix s from B is significant, we test if $A^I \cap C^s(s)$ is different from A^I. This requires computation of order $O(n)$.

Appendix D

Additional Experimental Observations

D.1 Base frequency in suffix selection

In section 4.8.3 it is discussed that bases that occur frequently are more likely to be valid, and hence the morphological extensions in decompositions involving more frequent bases are more likely to be valid. In table D.1 we summarize the effect of base frequency on the selection of morphological extensions. The results are contrary to our expectations. In table D.2 we put an additional restriction, namely, the bases should have at least two phonemes in it. Though there is a marginal improvement in the results, there is no significant effect of base frequency.

Total number of distinct words 20140
Actual number of suffixes present 187

Base frequency threshold	B	C	Q	S	Precision (%)	Recall (%)
2	10615	2184	657	182	1 69	97 33
3	10558	2132	651	181	1 69	96 79
4	10484	2076	639	181	1 70	96 79
5	10406	2015	627	179	1 69	95 72
6	10333	1946	616	179	1 70	95 72
7	10274	1894	596	179	1 71	95 72
8	10209	1860	588	178	1 71	95 19
9	10158	1802	575	177	1 71	94 65
10	10054	1747	556	174	1 70	93 05
11	10013	1693	539	172	1 69	91 98
12	9927	1644	525	170	1 68	90 91
13	9854	1589	512	168	1 68	89 84
14	9773	1544	494	167	1 68	89 30
15	9736	1483	482	161	1 63	86 10
16	9694	1437	466	159	1 61	85 03
17	9618	1399	446	159	1 63	85 03
18	9529	1364	445	156	1 61	83 42
19	9472	1340	439	155	1 61	82 89
20	9387	1303	428	152	1 59	81 28
21	9343	1298	426	152	1 60	81 28
22	9262	1277	418	150	1 59	80 21
23	9210	1234	413	150	1 60	80 21
24	9175	1203	408	149	1 60	79 68
25	9094	1186	400	148	1 60	79 14
26	9066	1167	393	148	1 61	79 14
27	9006	1112	382	146	1 60	78 07
28	8963	1097	370	145	1 59	77 54
29	8873	1092	364	144	1 60	77 01
30	8846	1080	360	143	1 59	76 47
31	8824	1061	350	141	1 57	75 40
32	8752	1030	329	138	1 55	73 80
33	8730	1030	328	138	1 56	73 80
34	8725	1005	323	138	1 56	73 80
35	8690	987	320	135	1 53	72 19
36	8569	985	313	134	1 54	71 66
37	8567	981	306	133	1 53	71 12
38	8542	952	299	131	1 51	70 05
39	8461	926	293	130	1 51	69 52
40	8443	925	292	129	1 50	68 98

S Suffix, Q Suffix-sequence, C Compound parts
B Invalid morphological extension

Table D 1 Effect of base frequency (without base length restriction) in selecting valid suffixes

Total number of distinct words 20140
Actual number of suffixes present 187

Base frequency threshold	B	C	Q	S	Precision (%)	Recall (%)
2	3187	1954	622	177	5 26	94 65
3	3122	1900	616	175	5 31	93 58
4	3028	1845	604	175	5 46	93 58
5	2938	1781	588	173	5 56	92 51
6	2853	1707	576	171	5 65	91 44
7	2786	1652	555	170	5 75	90 91
8	2711	1615	547	169	5 87	90 37
9	2652	1552	533	167	5 92	89 30
10	2545	1495	511	163	6 02	87 17
11	2501	1440	494	160	6 01	85 56
12	2412	1389	480	156	6 07	83 42
13	2335	1329	465	154	6 19	82 35
14	2252	1285	445	151	6 28	80 75
15	2218	1222	431	143	6 06	76 47
16	2163	1174	414	139	6 04	74 33
17	2095	1133	393	137	6 14	73 26
18	2000	1094	390	134	6 28	71 66
19	1967	1070	382	133	6 33	71 12
20	1879	1030	370	128	6 38	68 45
21	1824	1025	368	126	6 46	67 38
22	1756	1001	359	124	6 60	66 31
23	1696	954	354	124	6 81	66 31
24	1655	922	349	123	6 92	65 78
25	1583	903	340	122	7 16	65 24
26	1555	880	329	121	7 22	64 71
27	1510	824	317	119	7 31	63 64
28	1459	809	304	118	7 48	63 10
29	1398	805	298	116	7 66	62 03
30	1371	793	293	115	7 74	61 50
31	1343	776	284	112	7 70	59 89
32	1287	745	262	108	7 74	57 75
33	1264	745	260	108	7 87	57 75
34	1259	720	248	105	7 70	56 15
35	1220	701	243	102	7 72	54 55
36	1146	699	237	101	8 10	54 01
37	1144	695	230	100	8 04	53 48
38	1116	664	222	98	8 07	52 41
39	1062	638	216	94	8 13	50 27
40	1039	636	210	90	7 97	48 13
41	1018	614	205	87	7 87	46 52
42	962	609	205	87	8 29	46 52
43	923	609	204	87	8 61	46 52
44	867	603	201	86	9 02	45 99
45	865	580	177	84	8 85	44 92
46	850	575	176	84	8 99	44 92
47	850	567	165	82	8 80	43 85
48	719	539	162	79	9 90	42 25
49	719	539	162	79	9 90	42 25
50	719	539	162	79	9 90	42 25
51	680	518	150	72	9 57	38 50
52	680	518	150	72	9 57	38 50

S Suffix, Q Suffix-sequence, C Compound parts,
B Invalid morphological extension

Table D 2 Effect of base frequency (bases with two or more phonemes) in selecting valid suffixes

Bibliography

[1] James Allen. *Natural Language Understanding.* The Benjamin/Cummings Publishing Company Inc., Redwood City, second edition, 1995.

[2] Hemchandra Baruah. *Hem Kosha.* Hemkosh Prakashan, Guwahati, sixth edition, 1985.

[3] Satyanath Bora. *bahal byaakaran.* Jnananath Bora, Guwahati, 1968.

[4] Hagit Borer. Morphology and syntax. In Andrew Spencer & Arnold M Zwicky, editor, *The Handbook of Morphology*, pages 151–190. Blackwell Publishers Ltd., 1998.

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

[6] Eric Brill. Unsupervised learning of disambiguation rules for part of speech tagging. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 1–13. Association of Computational Linguistics, 1995.

[7] Andrew Carstairs-McCarthy. Phonological constraints on morphological rules. In Andrew Spencer & Arnold M Zwicky, editor, *The Handbook of Morphology*, pages 144–148. Blackwell Publishers Ltd., 1998.

[8] Tsong Yueh Chen, Fei-Ching Kuo, and Robert Merkel. On the statistical properties of the f-measure. In *Proceedings of the 4th International Conference on Quality Software (QSIC 2004)*, pages 146–153. IEEE, Sept 2004.

[9] Noam Chomsky. On cognitive structures and their development: A reply to piaget. In M. Piattelli-Palmarini, editor, *Languages and Learning: The Debate between Jean Piaget and Noam Chomsky.* Routledge and Kegan Paul, London, 1980.

[10] V J Cook and Mark Newson. *Chomsky's Universal Grammar An Introduction.* Blackwell Publishers Ltd., 108 Cowley Road, Oxford, UK, second edition, 1996.

[11] Thomas H Cormen, Charles E Leiserson, and Ronald L Rivest. *Introduction to Algorithms.* Prentice Hall of India, New Delhi, 1990.

[12] Mathias Creutz. Unsupervised segmentation of words using prior distributions of morph length and frequency. In *In Proceedings of ACL-03, the 41st Annual Meeting of the Association of Computational Linguistics*, pages 280–287, Sapporo, Japan, July 2003.

[13] Mathias Creutz and Krista Lagus. Induction of a simple morphology for highly-inflecting languages. In *Proceedings of the 7th Meeting of the ACL Special Interest Group in Computational Phonology (SIGPHON), Barcelona*, pages 43–51, July 2004.

[14] Mathias Creutz and Krista Lagus. Inducing the morphological lexicon of a natural language from unannotated text. In *In Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR'05)*, pages 106–113, June 2005.

[15] Walter Daelemans. Memory-based lexical acquisition and processing. In *{EAMT} Workshop*, pages 85–98, 1993.

[16] Éric Gaussier. Unsupervised learning of derivational morphology from inflectional lexicons. In *ACL '99 Workshop Proceedings: Unsupervised Learning in Natural Language Processing*, pages 24–30. ACL, 1999.

[17] Howard Gardner. Cognition comes of age (foreword). In M. Piattelli-Palmarini, editor, *Languages and Learning: The Debate between Jean Piaget and Noam Chomsky.* Routledge and Kegan Paul, London, 1980.

[18] Michael Gasser. Acquiring receptive morphology: A connectionist approach. In *Meeting of the Association for Computational Linguistics*, pages 279–286, 1994.

[19] John Goldsmith. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–193, 2001.

[20] Golokchandra Goswami. *asamiyaa byaakaranar moulik bisaar.* Bina Library, Guwahati, 1990.

[21] Peter A. Heeman. POS tagging versus classes in language modelling. In *Sixth Workshop on Very Large Corpora*, pages 179–187, Montreal, August 1998.

[22] Graeme Hirst. Lexical disambiguation. In *Semantic Interpretation and the Resolution of Ambiguity*, pages 77–95. Cambridge University Press, 1986.

[23] A S. Hornby. *Oxford Advanced Learner's Dictionary of Current English.* Oxford University Press, Oxford, fourth edition, 1989.

[24] Norbert Hornstein. *Move! A Minimalist Theory of Construal.* Blackwell Publishers Ltd., 108 Cowley Road, Oxford, UK, 2001.

[25] Lucja M Iwanska. Natural language is a powerful knowledge representation system: the uno model. In *Natural Language Processing and Knowledge Representation*, pages 7–64. University Press(India) Ltd., Hyderabad, 2001.

[26] Banikanta Kakati. *Assamese, Its Formation and Development.* LBS Publication, G.N.B. Road, Guwahati, fifth edition, 1995.

[27] Brian S Kernighan and Dennis M Ritchie. *The C Programming Language.* Prentice Hall of India, New Delhi, second edition, 1988.

[28] Rochelle Leiber. *Deconstructing morphology: Word formation in syntactic theory.* University of Chicago Press, Chicago, 1992.

[29] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2), 1993.

[30] Kaliram Medhi. *অসমীয়া ব্যাকৰণ আৰু ভাষাতত্ত্ব Assamese Grammar and Origin of the Assamese Language.* Lawyer's Book Stall, Guwahati, third edition, 1999.

[31] Andrei Mikheev. Automatic rule induction for unknown word guessing. *Computational Linguistics*, 23(3):405–423, 1997.

[32] Maheswar Neog and Upendranath Goswami. *Chandrakanta Abhidhan.* University Department of Publication, Gauhati University, Guwahati, third edition, 1988.

[33] Jean Piaget. The psychogenesis of knowledge and its epistemological significance. In M. Piattelli-Palmarini, editor, *Languages and Learning: The Debate between Jean Piaget and Noam Chomsky.* Routledge and Kegan Paul, London, 1980.

[34] M. Piattelli-Palmarini. Introduction: How hard is the hard core of a scientific program. In M. Piattelli-Palmarini, editor, *Languages and Learning: The Debate between Jean Piaget and Noam Chomsky*, pages 1–20. Routledge and Kegan Paul, London, 1980.

[35] M. Piattelli-Palmarini. *Languages and Learning: The Debate between Jean Piaget and Noam Chomsky.* Routledge and Kegan Paul, London, 1980.

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

[37] Gabor Proszeky and Balazs Kis. A unification-based approach to morpho-syntactic parsing of agglutinative and other (highly) inflectional languages. In *ACL'99 37th Annual Meeting of the Association of Computational Linguistics*, pages 261–268. Association for Computational Linguistics, June 1999.

[38] Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142. University of Pennsylvania, 1996.

[39] M Saravanan, P C Reghu Raj, Vadali Srinivasa Murty, and S Raman. Improved porter's algorithm for root word stemming. In *International*

Conference on Natural Language Processing, pages 21–30, Mumbai, December 2002.

[40] Durgashankar Dev Sarma. *sahaj byaakaran.* Assam State Textbook Production and Publication Corporation Ltd., Guwahati-1, 1977.

[41] Gerold Schneider. *An Introduction to Government and Binding.* University of Zurich, *http://www.ifi.unizh.ch/CL/gschneid/dreitaegig.ps.gz*, 1998.

[42] Utpal Sharma, Jugal Kalita, and Rajib Das. Classification of words based on affix evidence. In *International Conference on Natural Language Processing*, pages 31–39, Mumbai, December 2002.

[43] Utpal Sharma, Jugal Kalita, and Rajib Das. Root word stemming by multiple evidence from corpus. In *6th International Conference on Computational Intelligence and Natural Computing, CINC-2003*, North Carolina, September 2003.

[44] Matthew G Snover, Gaja E Jarosz, and Michael R Brent. Unsupervised learning of morphology using a novel directed search algorithm: Taking the first step. In *Workshop on Morphological and Phonological Learning, ACL-2002*, pages 11–20, Philadelphia, July 2002.

[45] Brants Thorsten. TnT – a statistical part-of-speech tagger. In *Proceedings of the 6th Applied NLP Conference, ANLP-2000*, Seattle, WA, 2000.

[46] Antal Bosch van den and Walter Daelemans. Memory-based morphological analysis. In *Proceedings of the 37th Annual Meeting of the ACL*, pages 285–292. University of Maryland, 1999.

[47] Srisa Chandra Vasu. *The Ashtadhyayi of Panini (Edited and Translated into English)*, volume I. Motilal Banarsidass, Delhi, 1891.

Index