North East University Bangladesh

Department of Computer Science and Engineering



**Unsupervised Bangla Word Segmentation Using Transitional Probability**

**By**

|  |  |  |
| --- | --- | --- |
|  | Ayon Dey  Reg. No: 160103020015  BSc(Engg) in CSE  4th year 3rd semester |  |

**Supervised By**

Mir Lutfur Rahman

Lecturer

Department of Computer Science and Engineering

16th January, 2019

**Unsupervised Bangla Word Segmentation Using Transitional Probability**



A Thesis submitted to the Department of Computer Science and Engineering,

North East University Bangladesh, in partial fulfillment of the requirements  
for the degree of Bachelor of Science in Computer Science and Engineering

**By**

|  |  |  |
| --- | --- | --- |
|  | Ayon Dey  Reg. No: 160103020015  BSc(Engg) in CSE  4th year 3rd semester |  |

**Supervised By**

Mir Lutfur Rahman

Lecturer

Department of Computer Science and Engineering

16th January, 2019

**Recommendation Letter from Thesis Supervisor**

This Student, *Ayon Dey*, whose thesis entitled *“Unsupervised Bangla Word Segmentation Using Transitional Probability”,* is under my supervision and agrees to submit for examination.

Signature of the Supervisor ：

Mir Lutfur Rahman

Lecturer

Department of Computer Science and Engineering

North East University Bangladesh

**Qualification Form of BSc (Engg) Degree**

Student Name： Ayon Dey

Thesis Title： Unsupervised Bangla Word Segmentation Using Transitional Probability

This is to certify that the thesis is submitted by the student named above in January 16, 2019. It is qualified and approved by the following persons and committee.

|  |  |
| --- | --- |
| **Head of the Dept.**  Noushad Sojib  Assistant Professor  Department of CSE  North East University Bangladesh | **Supervisor**  Mir Lutfur Rahman  Lecturer  Department of CSE  North East University Bangladesh |

## Acknowledgement

At first, I thank to the almighty for always being there with me. I also thank to my family members for supporting me all the way. I would like to thank, Al Mehdi Sadaat Chowdhury Sir for helping me to finding my research field. I thank Mir Lutfur Rahman Sir for his support throughout the thesis. It was only through his encouragement and support that I did this work.

Special thanks to Saber Vai and Mr. Dip Dasgupta for helping me collecting the data.

# Abstract

Word Segmentation is the process of segmenting the word into prefix, root and suffix that is helpful in the speech signal. It is the process of finding out the smallest meaning bearing elements of natural languages. In this research, We described segmenting the word in an unsupervised approach by finding words that appears as substring of other words using trie data structure and detecting changes using transitional probability. I use the unsupervised approach because it results with little extra manual effort. I will describe the two approaches that work together to finding out the morphemes.1) Finding morphemes that appear as substring of other string. 2) Calculating Transitional Probability. This algorithm gives me good result according to its simplicity. It is evaluated on a set of 1575 human segmented Bangla words, the 432-line python program achieved an accuracy of **80.95%.**

**Keywords:** Word Segmentation, Morpheme Induction, Transitional Probability, Forward Trie, Backward Trie, Pruning, Segmenting, Suffix, Prefix, Root.

# Table of Contents

[Abstract i](#_Toc30194150)

[Table of Contents ii](#_Toc30194151)

[List of Figures v](#_Toc30194152)

[List of Tables vi](#_Toc30194153)

[INTRODUCTION 1](#_Toc30194154)

[1.1 Natural Language Processing 1](#_Toc30194155)

[1.1.1 What is NLP used for? 2](#_Toc30194156)

[1.1.2 How Does NLP Works? 2](#_Toc30194157)

[1.1.3 What are the techniques used in NLP? 3](#_Toc30194158)

[1.1.3.1 Syntax 3](#_Toc30194159)

[1.1.3.2 Semantics 3](#_Toc30194160)

[1.2 Trie 4](#_Toc30194161)

[1.2.1 Search in a trie 4](#_Toc30194162)

[1.2.2 Insertion in a Trie 6](#_Toc30194163)

[1.2.3 Performance of Trie 7](#_Toc30194164)

[1.3 Probability 7](#_Toc30194165)

[1.3.1 Conditional Probability 8](#_Toc30194166)

[1.3.2 Transition Probability 8](#_Toc30194167)

[1.4 Outline of the report 9](#_Toc30194168)

[BACKGROUND STUDY 10](#_Toc30194169)

[2.1 Minimally supervised Morphological Analysis by Multimodal Alignment 10](#_Toc30194170)

[2.2 Morphemes as Necessary Concept for Structures Discovery from Untagged Corpora 11](#_Toc30194171)

[2.3 Inflectional Morphology Synthesis for Bengali Noun, Pronoun and Verb Systems. 11](#_Toc30194172)

[2.4 A Simpler, Intuitive Approach to Morpheme Induction. 11](#_Toc30194173)

[2.5 A Non-Dictionary-Based Thai Word Segmentation Using Decision Trees. 12](#_Toc30194174)

[2.6 Unsupervised Word Segmentation in Bangla 13](#_Toc30194175)

[2.7 Feature Unification for Morphological Parsing 14](#_Toc30194176)

[2.8 Transition based Neural Word Segmentation 15](#_Toc30194177)

[2.9 Unsupervised Word Segmentation and Lexicon Discovery using acoustic word embeddings. 15](#_Toc30194178)

[2.10 Transitional Probability and Word Segmentation 16](#_Toc30194179)

[2.11 Summary of the Literature: 16](#_Toc30194180)

[**Chapter 3** 17](#_Toc30194183)

[METHODOLOGY 17](#_Toc30194184)

[3.1 Collection of Data 17](#_Toc30194185)

[3.1.1 Pre-Processing: 18](#_Toc30194186)

[3.2 Methodology 19](#_Toc30194187)

[3.2.1 Building the Lexicographic Tries 19](#_Toc30194188)

[3.2.2 Scoring Potential Morphemes 20](#_Toc30194189)

[3.2.3 Pruning 21](#_Toc30194190)

[3.2.4 Segmenting 21](#_Toc30194191)

[RESULTS AND DISCUSSION 22](#_Toc30194192)

[4.1 Future Works 23](#_Toc30194193)

[CONCLUSION 24](#_Toc30194194)

[References 25](#_Toc30194195)

# List of Figures

**Figure 1** Creating A Trie. 4

**Figure 2** Finding “shells” in the trie 5

**Figure 3** Finding “She” in Trie. 5

**Figure 4** Finding “shell” in Trie. 6

**Figure 5** Finding “shelter” in Trie. 6

**Figure 6** Inserting “shore” in Trie. 7

**Figure 7**: Recall, Precision and Accuracy 13

**Figure 8** Sample of Unique Bengali word Dataset 17

**Figure 9** Sample data from Bengali corpus-2 18

**Figure 10** A Sample Dataset after preprocessing 19

# List of Tables

**Table 1** Performance of combined alignment models on 4 classes of past-tense English verbs

. 10

**Table 2** Top English Morphemes 12

**Table 3** Resource Usages of Different Test Data. 12

**Table 4** Evaluation results of Reports. 12

**Table 5** Exact accuracy,Precision,Recall and F-Score of unsupervised word segmentation in Bangla. 14

**Table 6** Sample of Bengali Stop-words 18

**Table 7** Datasets details after preprocessing 18

**Table 8** Top Score Suffix List 22

**Table 9** Top Score Prefix List 22

**Chapter 1**

# INTRODUCTION

Bengali are the official and most widely spoken language of Bangladesh and second most widely spoken language of India. It is an Indo-Aryan language which has 260-300 million speakers worldwide. Bengali is the 7th most spoken language by total number of speakers in the world. Though it is very popular language, there are very little significant research in computational Bengali linguistics has been done. Enhancement of Computational linguistics can help to increase human and computer interaction. Segmentation is the process of segmenting the word into prefix, root and suffix by means of marks in the speech signal. It is the process of finding out the smallest meaning bearing elements of natural languages.

Unsupervised Word Segmentation is the process of segmenting the word into prefix, root and suffix without prior knowledge of language specific rules. Supervised methods have reported great results for word segmentation, but their applicability is limited due to their dependence on human efforts and time consuming. For word segmentation, unsupervised methods are has great interest because of, they can learn to perform accurate word segmentation given input of any human language with little extra manual effort. They may give computational explanation on how children segment speech and discover words, starting from a state where they don’t know any word knowledge.

The key idea of my research is to use words that appear as substrings of other words and transitional probabilities together to detect morpheme boundaries.

The left of this chapter is about summary of NLP (Natural Language Processing), Trie Data Structure, Transitional Probability which should be understood for my approach of word segmentation.

## Natural Language Processing

Natural Language Processing is the technology used to aid computers to understand the human’s natural language.

Natural Language Processing, usually shortened as NLP, is a branch of artificial intelligence that deals with the interaction between computers and humans using the natural language.

The ultimate objective of NLP is to read, decipher, understand, and make sense of the human languages in a manner that is valuable.

Most NLP techniques rely on machine learning to derive meaning from human languages.

In fact a typical interaction between humans and machines using Natural Language Processing could go as follows:

1. A human talks to the machine
2. The machine captures the audio
3. Audio to text conversion takes place
4. Processing of the text’s data
5. Data to audio conversion takes place
6. The machine responds to the human by playing the audio file.

### What is NLP used for?

Natural Language processing is the driving force behind the following common applications:

1. Language translation applications such as Google Translate.
2. Word Processors such as Microsoft Word and Grammarly that employ NLP to check grammatical accuracy of texts.
3. Interactive Voice Response (IVR) applications used in call centers to respond to certain user’s requests.
4. Personal assistant applications such as OK Google, Siri, Cortana and Alexa.

### How Does NLP Works?

Natural Language Processing entails applying algorithms to identify and extract the natural language rules such that the unstructured language data is converted into a form that computers can understand.

When the texts have been provided, the computer will utilize algorithms to extract meaning associated with every sentence and collect the essential data from them.

Sometime, the computer may fail to understand the meaning of a sentence well, leading to Obscure results.

For example, a humorous incident occurred in the 1950s [15] during the translation of some words between the English and the Russian languages.

Here is the biblical sentence that required translation:

“The spirit is willing, but the flesh is weak.”

Here is the result when the sentence was translated to Russian and back to English:

“The vodka is good, but the meat is rotten.”

### What are the techniques used in NLP?

Syntactic analysis and semantic analysis are the main techniques used to complete Natural Language Processing tasks.

Here is a description on how they can be used:

#### Syntax

Syntax refers to the arrangement of words in a sentence such that they make grammatical sense.

In NLP, syntactic analysis is used to assess how the natural language aligns with the grammatical rules.

Computer algorithms are used to apply grammatical rules to a group of words and derive meaning from them.

Here are some syntax techniques that can be used:

* **Lemmatization**: It entails reducing the various inflected forms of a word into a single form for easy analysis.
* **Morphological** **segmentation/Word Segmentation**: It involves dividing words into individual units called morphemes.
* **Text segmentation**: It involves dividing a large piece of continuous text into distinct units.
* **Part-of-speech tagging**: It involves identifying the part of speech for every word.
* **Parsing**: It involves undertaking grammatical analysis for the provided sentence.
* **Sentence breaking**: It involves placing sentence boundaries on a large piece of text.
* **Stemming**: It involves cutting the inflected words to their root form.

#### Semantics

Semantics refers to the meaning that is conveyed by a text. Semantic analysis is one of the difficult aspects of Natural Language Processing that has not been fully resolved yet.

It involves applying computer algorithms to understand the meaning and interpretation of words and how sentences are structured.

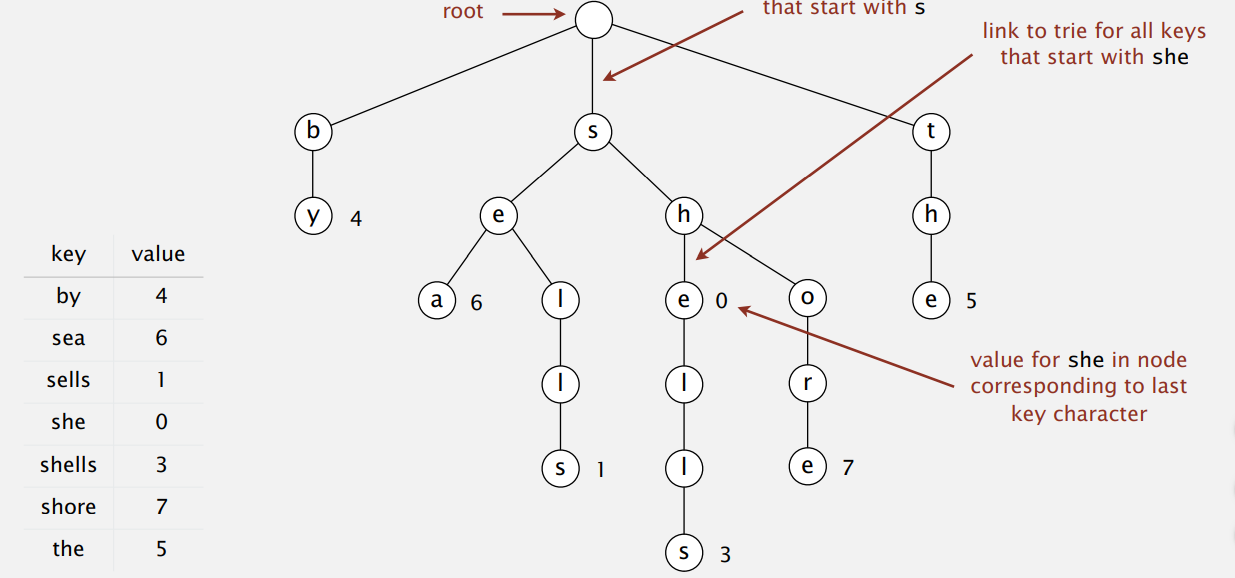
Here are some techniques in semantic analysis:

* **Named entity recognition (NER):** It involves determining the parts of a text that can be identified and categorized into preset groups. Examples of such groups include names of people and names of places.
* **Word sense disambiguation:** It involves giving meaning to a word based on the context.
* **Natural language generation**: It involves using databases to derive semantic intentions and convert them into human language.

## Trie

A trie (pronounced “try”) is a tree representing a collection of strings with one node per common prefix. Smallest tree such that:

* Store characters in nodes(not keys)
* Each node has R children ,one for each possible character.(For now I do not draw null links)



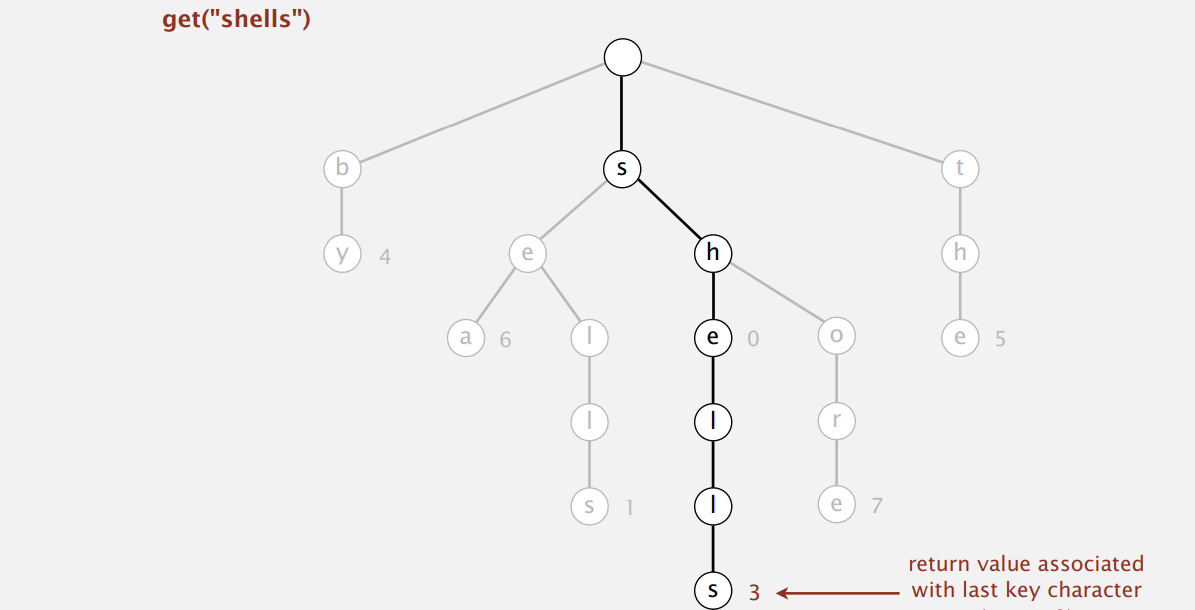
**Figure 1** Creating A Trie [13]

### 1.2.1 Search in a trie

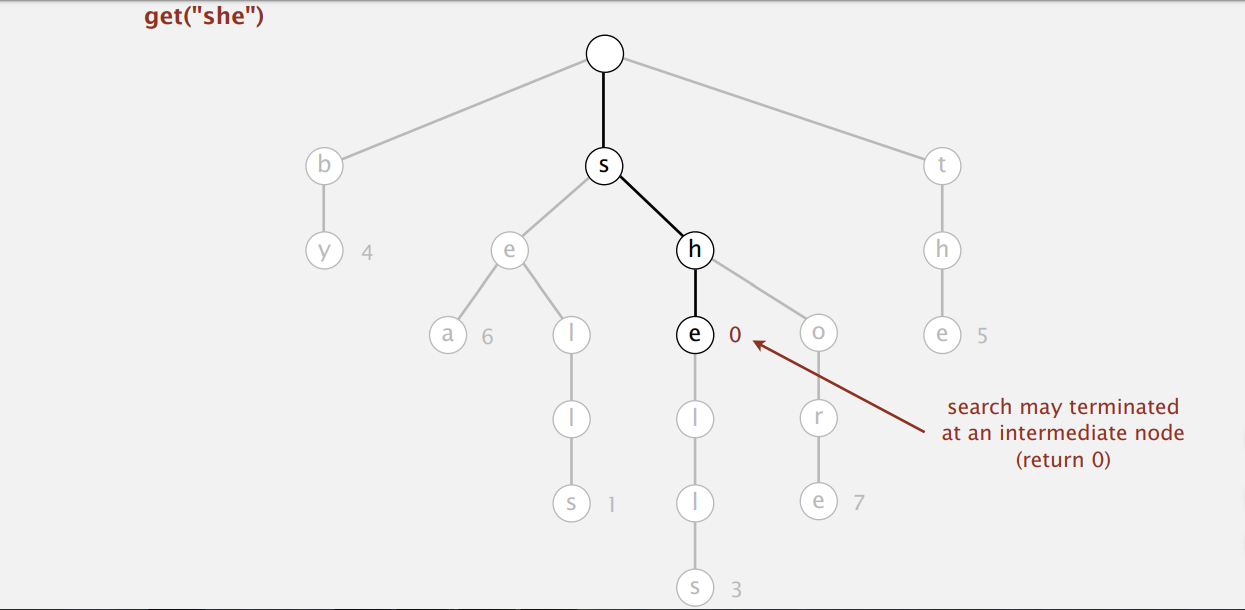
Follow links corresponding to each character in the key.

**Search hit:** node where search ends has a non-null value.

Here Figure 2 is for searching “shells” and Figure 3 is for searching “she”.

****

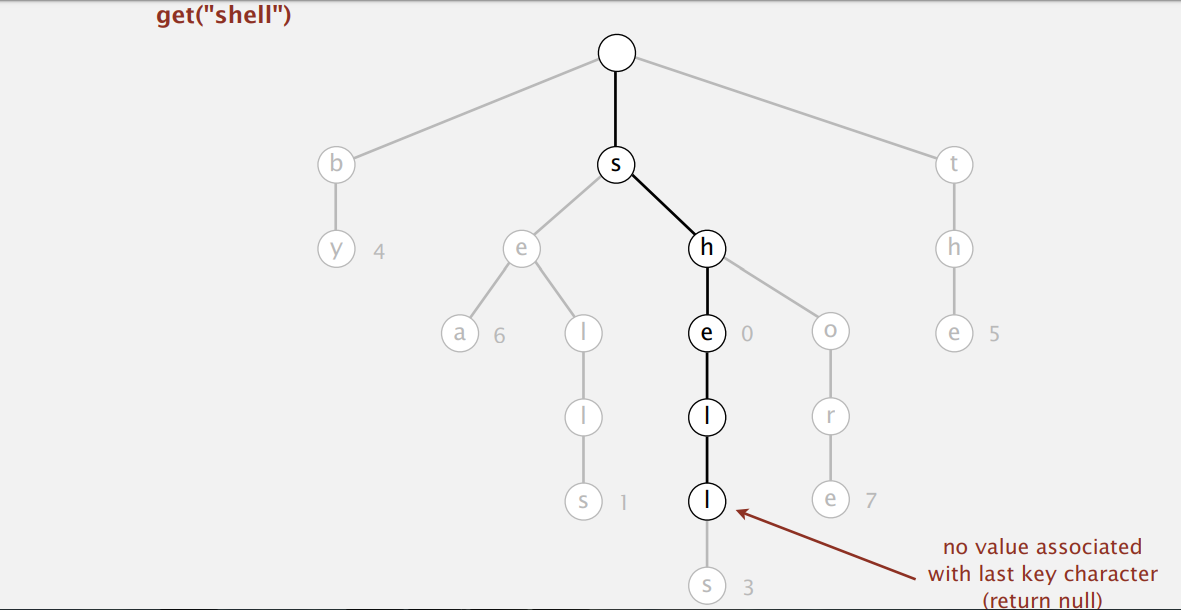
**Figure 2** Finding “shells” in the trie [13]

****

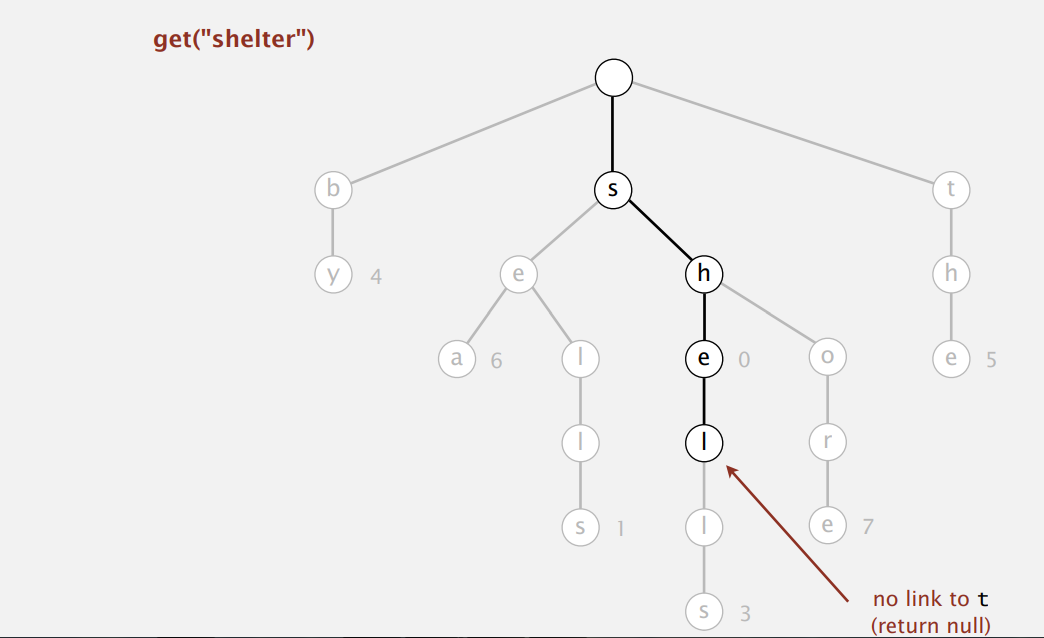
**Figure 3** Finding “She” in Trie [13]

**Search miss:** Reach null link or node where search ends has null value.

Here Figure 4 is for searching “shell” and Figure 5 is for searching “shelter”

****

**Figure 4** Finding “shell” in Trie [13]

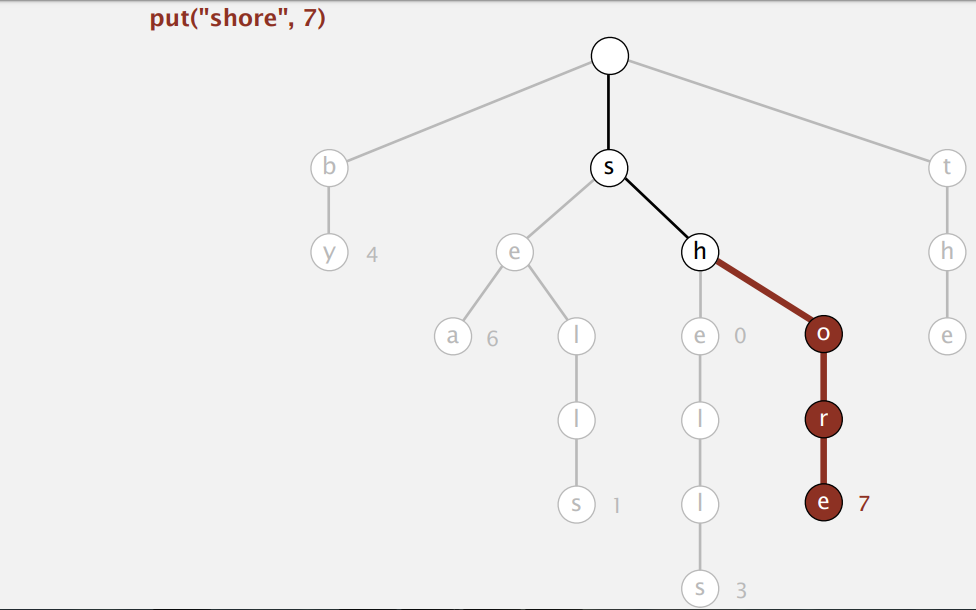
****

**Figure 5** Finding “shelter” in Trie[13]

### 1.2.2 Insertion in a Trie

Follow links corresponding to each character in the key.

* Encounter a null line: Create new node.
* Encounter the last character of the key: set value in that node.



**Figure 6** Inserting “shore” in Trie[13]

### 1.2.3 Performance of Trie

Trie is an efficient information retrieval data structure. Using Trie, search complexities can be brought to optimal limit(key Length).If we store keys in binary search tree, a well balanced BST will need time proportional to M\*log N ,where M is maximum string length and N is number of keys in tree. Using Trie, we can search the key in O (M) time.

Insert and search costs O (key length), however the memory requirements of Trie is O (Alphabet Size\*key\_length\*N) where N is number of keys In Trie.

## Probability

Probability is the likelihood or chance of an event occurring

For example, the probability of flipping a coin and it being heads is ½, because there is 1 way of getting a head and the total number of possible outcomes is 2 (a head or tail). We write P (heads) = ½.

* The probability of something which is certain to happen is 1.
* The probability of something which is impossible to happen is 0.
* The probability of something not happening is 1 minus the probability that it will happen.

### 1.3.1 Conditional Probability

The conditional probability of an event B is the probability that the event will occur given the knowledge that an event A has already occurred. This probability is written P (B|A), notation for the probability of B given A. In the case where events A and B are independent (where event A has no effect on the probability of event B), the conditional probability of event B given event A is simply the probability of event B, that is P(B).

If events A and B are not independent, then the probability of the intersection of A and B (the probability that both events occur) is defined by,

P (A and B) = P (A) P (B|A).

From this definition, the conditional probability P (B|A) is easily obtained by dividing by P (A):

In a card game, suppose a player needs to draw two cards of the same suit in order to win. Of the 52 cards, there are 13 cards in each suit. Suppose first the player draws a heart. Now the player wishes to draw a second heart. Since one heart has already been chosen, there are now 12 hearts remaining in a deck of 51 cards. So the conditional probability P (Draw second heart|First card a heart) = 12/51.

Suppose an individual applying to a college determines that he has an 80% chance of being accepted, and he knows that dormitory housing will only be provided for 60% of all of the accepted students. The chance of the student being accepted and receiving dormitory housing is defined,  
P(Accepted and Dormitory Housing) = P(Dormitory Housing|Accepted)P(Accepted) = (0.60)\*(0.80) = 0.48.

### 1.3.2 Transition Probability

The one-step transition probability is the probability of transitioning from one state to another in a single step. The Markov chain is said to be time homogeneous if the transition probabilities from one state to another are independent of time index.

Pij=Pr{Xn=j|Xn-1=i}

The transition probability matrix, P, is the matrix consisting of the one-step transition probabilities Pij,

The m-step transition probability is the probability of transitioning from state i to state j in m steps.

Pij (m) =Pr {Xn+m=j|Xn=i}

The m-step transition matrix whose elements are the m-step transition probabilities Pij (m) is denoted as P (m).

The m-step transition probabilities can be found from the single-step transition probabilities as follows.

To transition from i to j in m steps, the process can first transition from i to r in m-k steps, and then transition from r to j in k steps, where 0<k<m.

∏(m)=∏(0)Pm  
Where ∏(0)is the vector containing the initial probabilities of being in each state at time 0.

**Example:**

Suppose a word is “ছাগল”

Transitional Probability=Frequency of words (ছাগল)/Frequency of words (“ছাগ”)

## Outline of the report

The first chapter of this report introduced the basics of word segmentation, trie data structure, conditional probability, transitional probability which is required for my approach.

In second chapter of this report, some articles related to word segmentation/morpheme induction have been summarized. This articles elaborates some key approach for word

Segmentation in English, Chinese, Bangla.

In third chapter of this report, I have described the preprocessing step for collecting Bengali dataset as well as the methodology of this work.

In the fourth chapter of this report the result of this work is described.

In the last chapter of this report, I have summarized my work. I described the accuracy as well as the whole process.

**Chapter 2**

# BACKGROUND STUDY

There has been a good storage of word segmentation research all over the world. Among them I have chosen some good research from English, Thai, Chinese and Bengali Language.

This chapter takes some snapshot of relevant works already been done for word segmentation.

## Minimally supervised Morphological Analysis by Multimodal Alignment

In research [1] Yarowsky and Wicentowski presented a corpus-based algorithm capable of inducing inflectional morphological analysis of both regular and highly irregular forms (such as brought->bring).It is useful to consider this task as three separate steps:

Estimate a probabilistic alignment between inflected forms and root forms in a given language.

Train a supervised morphological analysis learner on a weighted subset of these aligned pairs.

Use the result of step 2 as either a standalone analyzer or a probabilistic scoring component to iteratively refine the alignment in step 1.

Performance after testing is given below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Combination of Similarity Models** | **# of Iterations** | **All Words(3888)** | **Highly Irregular(128)** | **Simple Concat.(1877)** | **Non-Concat.(1883)** |
| FS(Frequency Sim) | (Iter 1) | 9.8 | 18.6 | 8.8 | 10.1 |
| LS(Levenshtein Sim) | (Iter 1) | 31.3 | 19.6 | 20.0 | 34.4 |
| CS(Context Sim) | (Iter 1) | 28.0 | 32.8 | 30.0 | 25.8 |
| CS+FS | (Iter 1) | 32.5 | 64.8 | 32.0 | 30.7 |
| CS+FS+LS | (Iter 1) | 71.6 | 76.5 | 71.1 | 71.9 |
| CS+FS+LS+MS | (Iter 1) | 96.5 | 74.0 | 97.3 | 97.4 |
| CS+FS+LS+MS | (Convg) | 99.2 | 80.4 | 99.9 | 99.7 |

**Table 1** Performance of combined alignment models on 4 classes of past-tense English verbs

Accuracy of the induced analysis of 3888 past tense test cases in English exceeds 99.2% for the set, with currently over 80% accuracy on the most highly irregular forms and 99.7% accuracy on forms exhibiting non-concatenative suffixation.

## Morphemes as Necessary Concept for Structures Discovery from Untagged Corpora

In research [2] Dejean presented an overview of a method which allows discovery of syntactic structures from untagged corpora. Three steps-

* The discovery of the most frequent morphemes of the language.
* The discovery of the other morphemes.
* The segmentation of the words of the corpus.

In case of result, the author check on the segmentation of 500 words randomly selected and they obtain 8 segmentations as wrong.

## Inflectional Morphology Synthesis for Bengali Noun, Pronoun and Verb Systems.

In research of [3] Bhattacharya, Coudhury, Sarkar, Basu presented a rule based approach for Bengali Morphological Synthesis. For nouns and pronouns, the synthesis engine selects the appropriate suffixes, determines the ordering among them and concatenates them according to the rules. The verb roots have been classified under 19 categories based on the syllable structure. A Finite State Machine is used to recognize the category of the root. Thereafter, a simple algorithm generates the inflected form of the verb based on suffix table and rule tables for the classes.

## A Simpler, Intuitive Approach to Morpheme Induction.

This research [4] keshava and Pitler presented a simpler, psychologically plausible algorithm to perform unsupervised learning of morphemes. There are two steps –

* Finding words that appear as substring of other words and
* Detecting changes in transitional probabilities.

The algorithm which is used in segmentation follows 4 basic steps-

* Build trees with probabilities based on corpus.
* Score word fragments using these trees to obtain a large list of morphemes.
* Prune this list of morphemes and
* Segment the test words using the morpheme list and the lexicographic trees.

This algorithm yields particularly good results given its simplicity and conciseness: evaluated on a set of 532 human segmented English Words, A 252-line program achieved an F-score of 80.92 %( Precision: 82.84%, Recall: 79.10%)

Some top Score morphemes (both suffix list and prefix list):

|  |  |
| --- | --- |
| **Morpheme** | **Score** |
| Un | 15858 |
| Re | 5312 |
| Dis | 3783 |
| Non | 2998 |
| Over | 2717 |
| Mis | 1812 |
| Ln | 1689 |
| Sub | 1632 |
| Pre | 1418 |
| Inter | 1189 |
| S | 24351 |
| Ly | 18847 |
| Ness | 10430 |
| Ing | 8740 |
| Ed | 5669 |
| Al | 2655 |
| Ism | 2169 |
| Less | 1940 |
| Ist | 1669 |
| Able | 1613 |

**Table 2:** Top English Morphemes

Time and space complexity:

|  |  |  |
| --- | --- | --- |
| **Words** | **Time** | **Space** |
| 532 | Om 27Sec | 139 MB |
| 167,377 | 34m 37Sec | 139 MB |

**Table 3:** Resource Usage of Different Test Data.

Accuracy of some other languages using same algorithm:

|  |  |  |  |
| --- | --- | --- | --- |
| **Language** | **Precision** | **Recall** | **F-Score** |
| Turkish | 72.68% | 43.01% | 54.04% |
| Finnish | 83.76% | 32.30% | 46.62% |

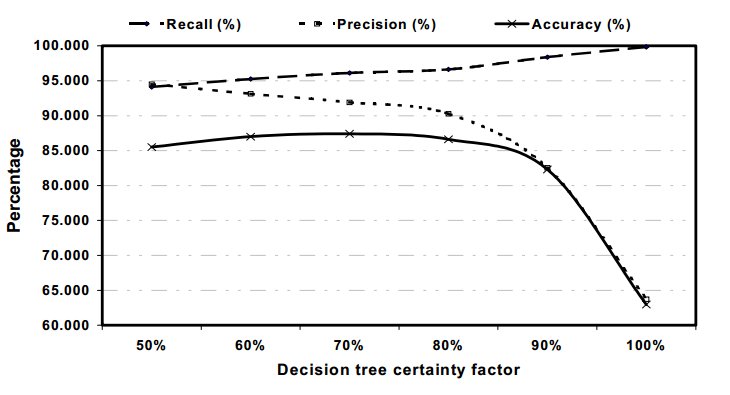
**Table 4:** Evaluation results of Reports

## A Non-Dictionary-Based Thai Word Segmentation Using Decision Trees.

If the dictionary is not sufficiently good, it will lead to a great number of unknown or unrecognized words. These unrecognized words certainly reduce segmentation accuracy. To solve such problem, they propose a method based on decision tree models. Without use of a dictionary, specific information, called syntactic attribute is applied on identify the structure of Thai Words.

In this research [5] Theeramunkong and Usanavasin presented they propose a word segmentation method that 1) uses a set of rules to combine contiguous characters to an inseparable unit(syllable-like unit) and 2) then applies a learned decision tree to combine these contiguous units to words.

They calculate the Precision, Recall and Accuracy as defined below:



**Figure 7**: Recall, Precision and Accuracy [5]

In our experiments, the best level of permission that leads to the highest accuracy is approximately equals to 70%, which gives the accuracy equal to87.41%, as shown in Figure 7.

## Unsupervised Word Segmentation in Bangla

This research [6] Dasgupta and V.Ng described unsupervised word segmentation in two steps-

* A morpheme induction step in which morphemes are automatically induced from a vocabulary consisting of words taken from a large ,unannoted corpus and
* A segmentation step in which a given word is segmented based on these automatically induced morphemes

In case of Morpheme induction they rely on a fairly simple for morpheme induction .Assume that A and B are two character sequences and AB and A are both found in the vocabulary, then they extract B as a candidate suffix. Similarly, if AB and B are both found in the vocabulary, then they extract A as a candidate prefix.

In case of Segmentation, given a word in the test set, they identified all possible segmentations of the word using only the induced affixes and roots. Then, they filter that candidate segmentation that violates any of the simple linguistic constraints below:

* There has to be at least one root in the segmentation.
* If a morpheme is a prefix, then the immediately following morpheme should be either a root or a prefix.
* If a morpheme is a suffix, then the immediately preceding morpheme should be either a root or a suffix.

Results-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System Variations** | **Exact Accuracy** | **Precision** | **Recall** | **F-Score** |
| Baseline(Linguistica) | 37.08 | 58.25 | 65.15 | 61.48 |
| Basic Induction | 46.67 | 76.66 | 66.2 | 71.04 |
| Composite Suffix Detection | 55.99 | 79.07 | 80.61 | 79.83 |
| Length Dependent Thresholds | 58.38 | 81.97 | 79.75 | 80.85 |
| Incorrect attachment Detection | 65.83 | 89.1 | 80.22 | 84.43 |

**Table 5:** Exact accuracy, Precision, Recall and F-Score of unsupervised word segmentation in Bangla.

## Feature Unification for Morphological Parsing

In this research [7] Dasgupta and Khan discussed about feature based morphological parsing for Bangla which gives us parts of speech and other morphological features in addition to the morpheme division.

There are two types of Morphological parser .They are –

Normal Morphological Parser which has 3 components.

**Lexicon**-The list of stems and affixes together with basic information about them (Whether a stem is a Noun stem or a Verb stem, etc)

**Morphotactics**-The model of morpheme ordering that explains which classes of morphemes can follow other classes of morphemes inside a word.

**Orthographic Rules**-These spelling rules are used to model the changes that occur in a word, usually when two morphemes combine, For example root word hat (হাট) is changed into het (হেট) when added with verb suffix to form a word hetechi (হেটেছি).

Feature Based Morphological Parsing-

Adding extra analytical component.

Create a parse tree and tokenize a word into morpheme and set it to the leaf node.

The reason behind they decide for a feature based morphological

Parsing-

The word grammar component can deduce the lexical category (part-of-speech) of a word.

The word grammar component offers a more powerful model of morph tactics.

Feature based morphology uses FSA more optimally.

The word grammar component can provide a full feature specification for a word.

## Transition based Neural Word Segmentation

In this research [8] M.Zhang, Y.Zhang, G.Fu described they studied a neural model for word-based Chinese word segmentation, by replacing the manually designed discrete features with neural features in a word-based segmentation framework. Experimental results demonstrate that word features lead to comparable performances to the best systems in the literature, and a further combination of discrete and neural features gives top accuracies.

## Unsupervised Word Segmentation and Lexicon Discovery using acoustic word embeddings.

In this research [9] Kamper, Jansen and Goldwater described a potential word segment (of arbitrary length) is embedded in a fixed-dimensional acoustic vector space. The model, implemented as a Gibbs Sampler, then builds a whole-word acoustic model in this space while jointly performing segmentation. They reported word error rates in a small-vocabulary connected digit recognition task by mapping the unsupervised decoded output to ground transcriptions. The model around 20% error rates, outperforming a previous HMM-based system by about 10% absolute. Moreover, in contrast to the baseline, our model does not require a pre-specified vocabulary size.

## Transitional Probability and Word Segmentation

In this research [10] Y.Xie aimed at reviewing the literature in the studies of the relationship between transitional probability and word segmentation in an attempt to emphasize statistical learning as the experience-dependent factor in language acquisition. Transitional probability, the crucial cue of the statistical relationship between syllables, is characterized by its tow computation directions: the forward transitional probability and backward transitional probability.

Transitional probabilities alone cannot generalize word segmentation in all linguistic input because of exceptions and variability in different languages. However, the facilitative role of

TPs should not be disregarded, as the experience with sound patterns in the early stage build the foundation for the lexical acquisition leading to meaningful communication. Moreover, transitional probabilities are often integrated with other cues like stress and co-articulation by infants to solve the problem of discovering word boundaries. Future research should  
explore more in cross-linguistic factors and the factors related with participants like adult second foreign language learners to shed new light on the topic of word segmentation.

## Summary of the Literature:

For this research work we have studied many research paper related to my research work. Among them I have selected 10 articles .From background we have observed there are very few works for word segmentation in Bangla. For supervised word segmentation we need to use rule based approach which is really time consuming because of collecting the huge case of rules as Bengali has a rich grammatical rules. Another reason for not choosing supervised approach because we have not any readymade huge segmented word list. For this purpose, A little extra manual effort, Unsupervised Word Segmentation may the best choice. In [4], Samara Keshava and Emily Pitler used a simple algorithm and got a good performance in English. They used forward Trie and Backward Trie along with Transitional Probability to segment the word into prefix, root and suffix. As Bengali is an Indo-European (it can be represented by trie data structure and also suitable for transitional probability) We used this simple algorithm. Hence, our target was to segment the word into its prefix, root and suffix in an unsupervised manner and increase the accuracy level.



**Chapter 3**

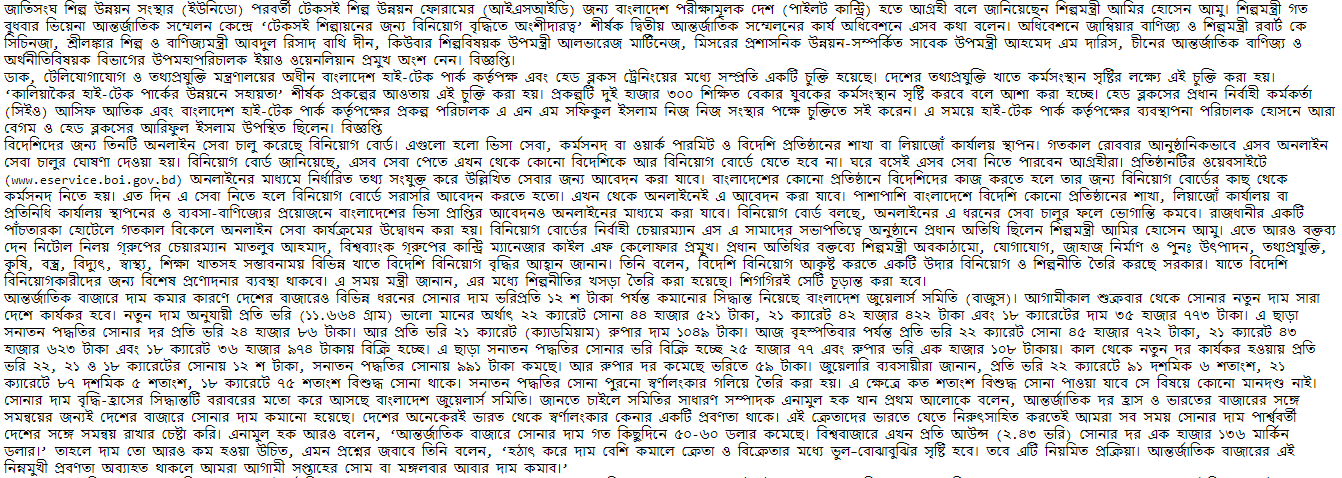
# METHODOLOGY

## Collection of Data

Collection of Data includes the data collection from various resources and processes them like crawling, indexing, filtering etc. which are used to collect the text that needs to be used to get the word segmentation, index them to store and retrieve in a better way. This is a very important stage of this research because a good dataset is necessary to lead better results. On this purpose dataset collected is around 400000 unique Bengali words from various dictionaries and a sample is given below. I also start collecting Bengali text from various fields like newspaper, articles, books etc. The corpus-1 is ‘prothom-alo 2017’ newspaper with 10, 55,550 sentences and 7, 06,374 unique words and other corpus-2 is created by combining different newspapers article is with 7, 87,252 sentences and 4, 79,386 unique words.



**Figure 8** Sample of Unique Bengali word Dataset[20]



**Figure 9** Sample data from Bengali corpus-2[19]

### 3.1.1 Pre-Processing:

Preprocessing is the very important step in natural language processing tasks. For word clustering it’s important to preprocessed data before used it in the model. The terms I used for preprocessing are:

* Whitespaces: [\s\u0020\u00a0\u180e\u202f\u205f\u3000\u2000-\u200a]
* English Char:[ a-z, A-Z]
* Digits:[0-9,\u09E6\u09E7\u09E8\u09E9\u09EA\u09EB\u09EC\u09ED\u09EE\u09EF ]
* Punctuations :[(),$%^&\*+={}\[\]:\"|\'\~`<>/,¦!?½£¶©¤¿º;-@?\_]
* Punctuations-sequences:['\"“”‘’]+|[.?!,…]+|[:;]

Stop-words: List of Bengali stop words.

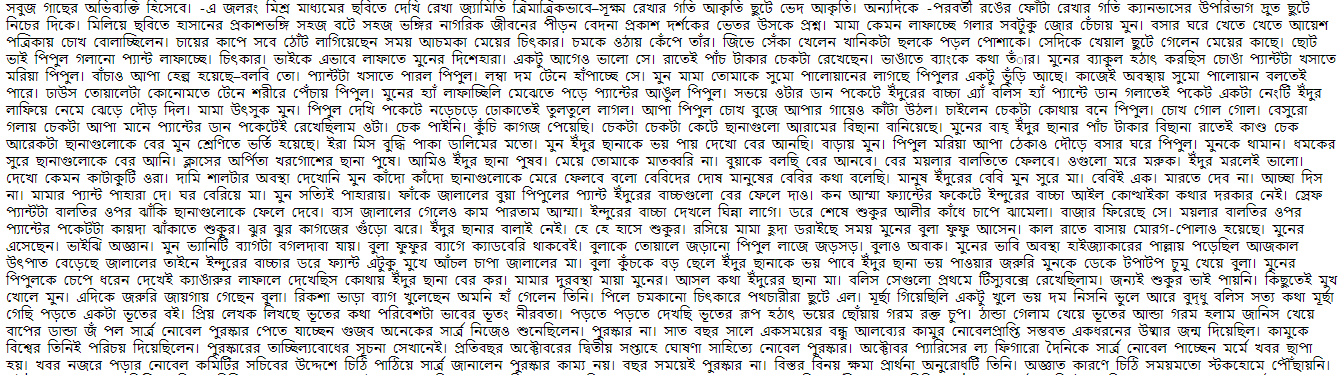
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| আমি | আর | এই | এ | পারে | যা |
| উপর | ইত্যাদি | অথচ | এটা | বরং | রাখা |
| এত | ঐ | কখনও | মতো | স্বয়ং | নেই |
| ও | তার | বা | হয় | মধ্যে | সব |

**Table 6**: Sample of Bengali Stop-words

After applying the preprocessing terms over the dataset, the dataset is cleaned up and ready for train the model and for testing also.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Corpus** | **File Type** | **Size** | **Total Sentences** | **Total Words** | **Total Unique Words** |
| Corpus-1 | TEXT | 2,30,626 KB | 10,60,065 | 1,34,63,128 | 4,37,791 |
| Corpus-2 | TEXT | 1,80,346 KB | 7,87,252 | 1,04,92,566 | 2,69,048 |

**Table 7**: Datasets details after preprocessing



**Figure 10** A Sample Dataset after preprocessing [18]

## Methodology

The complete methodology I have applied was tested on Intel® Core™ i5-4210U CPU @1.70 GHz 2.40 GHz processor with 4 GB of memory.

For Word Segmentation I will describe the two approaches that work together to detect morphemes. Firstly finding morphemes that appear as substrings of other words and then detecting changes in transitional probabilities. The algorithm has four basic steps-

* Build tries with probabilities based on the corpus.
* Score word fragments using these trees to obtain a large list of morphemes.
* Prune this list of morphemes and
* Segment the test words using the morpheme list and the lexicographic trees.

Each of these steps will be described:

### 3.2.1 Building the Lexicographic Tries

At the beginning of the algorithm, I created two trees of letters and their associated counts: the “forward trie” and “backward trie”. I explained here the construction of the “forward trie” (the other construction is symmetric). Suppose the alphabet of the language has *b* letters, and the longest word in the corpus consists of *d* letters. Then conceptually, i constructed a complete *b*-way trie with depth *d*. At each node, each of the *b* branches represents one of the letters in the language. Thus, any path from the root to some node spells out the starting fragment of some words, and the node itself contains the frequency of that string.The forward and backward tries allowed me to calculate conditional probabilities in O(1) time given a starting or ending substring of a word. For example, I used the forward trie to calculate *Prf* (র*|*ভালবাসা) (by dividing the frequency of words starting with “ভালবাসার” by the frequency of words starting with “ভালবাসা”). In the opposite direction, I used the backward trie to calculate *Prf* (আত্ম*|*রক্ষা) (by dividing the frequency of words ending in “আত্মরক্ষা” by the frequency of words ending in “রক্ষা”).

### 3.2.2 Scoring Potential Morphemes

Once i have finished constructing the tries as described above, I began finding morphemes. I  
maintained two lists of morphemes: a prefix list and a suffix list. To populate the suffix list, for each word, I scanned from the end of the word and considered every possible suffix in order of increasing length. Suppose I considered the suffix *Bβ* in the word *αABβ*. I hypothesized the proposed suffix is correct if,

* *αA* was also a word in the corpus,
* *Prf*(*A|α*) *≈* 1, and
* *Prf* (*B|αA*) *<*1.

Similarly, the criteria for determining if *αA* was a prefix in the word *αABβ* was as follows:

* *Bβ* was also a word in the corpus,
* *Prb*(*B|β*) *≈* 1, and
* *Prb* (*A|Bβ*) *<*1.

The first criterion corresponds to the observation that prefixes and suffixes were often added on to root words. For example, after removing the suffix “গুলো” from “জেলাগুলো”, the resulting fragment “জেলা” is still a word. The second and third criteria were checked using the forward and backward tries. They checked that the stem has multiple children (thus implying other prefixes or suffixes can be joined to the stem) and that the stem’s parent has only one child (thus identifying it as a true stem). Using the same example as before, the algorithm would check that *Prf* (া*|*জেল) *≈* 1, and that Prf (গুলো*|*জেলা) *<*1. If a given morpheme passed all three tests, I increased its score by 19 points; otherwise, decreased its score by 1. After i have iterated through the entire corpus, i consider all strings with positive scores morphemes.  
The rule of rewarding word fragments by 19 and punishing by 1 may seem arbitrary, but the constants were chosen so that a string has a positive final score only if it passes my tests at least five percent ( = 1/(1+19) ) of the times it appears. Moreover, the numbers 19 and 1 are not special; any positive numbers *x* and *y* such that *(y/(x+y))* = *.*05 would produce identical results. The rewarding and punishing scheme is more effective than checking the percentage of tests passed because given two morphemes with the same percentage, the more common morpheme will have a higher score. Thus, the punishing/rewarding scheme takes into account both the reliability and the frequency of the string appearing as a morpheme.

### 3.2.3 Pruning

Clearly, this method was not perfect. In particular, one problem that often arises that the final list of morphemes includes strings that are the concatenation of two other morphemes. For example, the list might include all of ‘গুলো’ ‘র’, and ‘গুলোর’. This is undesirable since the final step of segmenting words may process the word “জেলাগুলোর” as “জেলা+গুলোর”instead of as ‘জেলা+গুলো+র’. Fortunately, though, this problem has a relatively simple solution which i referred to as “pruning”. I scanned each list of morphemes, and if any morpheme was composed of two others with better scores, then it was thrown out.

### 3.2.4 Segmenting

Finally, I came to the actual segmenting of words. Given the list of morphemes, one possible approach was to simply peel morphemes off the ends of words as they were found. But there might be some problem. Thus, my method for segmenting was as follows.  
First, I scanned each word from the end, and find all morphemes *Bβ* from the suffix list such that my word could be written as *αBβ* (for some *α*). The morpheme with the lowest value of *Prf* (*B|α*) that was also smaller than 1 is chosen. If such a morpheme was found, it was removed and the processed was repeated until no more morphemes could be removed. I then repeated the same process, attempting to peel off morphemes in the prefix list from the beginning of the word (using *Prb*instead of *Prf*) morpheme.

An Overview of my implementation:

Creating Forward Trie and Backward Trie and inserting all the words



Counting the frequency of every possible substring of words.



Scoring the suffix list and prefix list considering transitional probability.



Storing the suffix and prefix which has positive value.



Segmenting the word based on suffix list and prefix list along with transitional Probability.



Prune the segmentation.



Calculate the accuracy comparing test segment set and predicted segment set.

**Chapter 4**

# RESULTS AND DISCUSSION

The algorithm described above was implemented as a python program. To determine the performance of the algorithm we ran our program on 1575 Human Segmented Words [21].

Our program identified a total of 4634 morphemes (2865 in the prefix list and 1769 in the suffix list). Table 8 contains the twelve highest-scoring morphemes from suffix list:

|  |  |
| --- | --- |
| **Morpheme** | **Score** |
| ের | 230551 |
| র | 164337 |
| ে | 147264 |
| কে | 136446 |
| ও | 135270 |
| ই | 104003 |
| সহ | 70787 |
| য় | 59262 |
| তে | 49261 |
| দের | 41676 |
| টি | 29517 |
| গুলো | 22738 |

**Table 8**: Top Score Suffix List

Table 9 contains the twelve highest-scoring morphemes from prefix list:

|  |  |
| --- | --- |
| **Morpheme** | **Score** |
| অ | 6891 |
| মহা | 6028 |
| গণ | 5411 |
| জন | 4720 |
| উপ | 4440 |
| আত্ম | 4023 |
| বিশ্ব | 3619 |
| শিল্প | 3226 |
| কর্ম | 3163 |
| শিশু | 2842 |
| শিক্ষা | 2837 |
| সু | 2693 |

**Table 9**: Top Score Prefix List

. Among 1575 words I found 1275 accurate segmentations which lead the accuracy of **80.95%.**

In [7] DasGupta and V.Ng tested their program in 2511 Human segmented words and got the accuracy of 65.83%.

In [4] Samarth Keshava and Emily Pitler evaluated their program on a set of 532 human segmented English Words and got the F-Score of 80.92%.

In [5] Theeramunkong and Usanavasin they got the accuracy of 63.72%.

My algorithm performs well given its conciseness and simplicity. The python implementation was a total of 432 lines including comments and the algorithm itself can be fully described. Example of words that our program segments correctly include “ধারাবাহিক”, “জীবনকাহিনী”, “সাগরের”,“আত্মরক্ষা”.First one was segmented in my code as “ধারা+বাহ+িক”,second one as “জীবন+কাহিনী”, third on as “সাগর+ের”, fourth one as “আত্ম+রক্ষা”

However, the algorithm is obviously not flawless. Consider a word such as “থিয়েটার” (with a correct segmentation of “থিয়েটার”).The letters “টার” always appear at the end of a word as a suffix. Thus, the potential morpheme ‘টার’ is rewarded far more frequently than it is punished and appears in the final list of morphemes. This omission causes me to incorrectly segment words such as “থিয়ে+টার”.

## Future Works

One notable feature of the algorithm is that it uses only a list of words and their frequencies. Clearly, contextual information is lost when Bangla text is collapsed into such list. We feel that the performance can be increased by considering the information about the word. Therefore, instead of feeding hundreds of thousands of words to the program at once, we could use deep neural network along with word2vec to consider the word in terms of contextual information. It would be interesting to compare my current results to those from the process.

**Chapter 5**

# CONCLUSION

There is no significant research for word segmentation in Bangla .In this research work we have already collected a large corpus of data which we already described. Made a forward trie and backward trie with this data and initially segment the word .But for removing error we used transitional probability along with pruning to minimize the error. Than we segmented the words using prefix and suffix list along with transitional probability. For accuracy checking we collected 1575 human segmented words and compare with the actual segmentation with my predicted segmentation. I got the accuracy of **80.95%.** In future, it would be grateful for us to increase the accuracy.

# 

# References

[1] Yarowsky, D., & Wicentowski, R. (2000, October). Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics* (pp. 207-216). Association for Computational Linguistics.

[2] Déjean, H. (1998, January). Morphemes as necessary concept for structures discovery from untagged corpora. In *Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning* (pp. 295-298). Association for Computational Linguistics.

[3] Bhattacharya, S., Choudhury, M., Sarkar, S., & Basu, A. (2005, March). Inflectional morphology synthesis for Bengali noun, pronoun and verb systems. In *Proc. of the National Conference on Computer Processing of Bangla (NCCPB 05)* (pp. 34-43).

[4] Keshava, S., & Pitler, E. (2006, April). A simpler, intuitive approach to morpheme induction. In *Proceedings of 2nd Pascal Challenges Workshop* (pp. 31-35).

[5] Theeramunkong, T., & Usanavasin, S. (2001, March). Non-dictionary-based Thai word segmentation using decision trees. In *Proceedings of the first international conference on Human language technology research* (pp. 1-5). Association for Computational Linguistics.

[6] Dasgupta, S., & Ng, V. (2007). Unsupervised word segmentation for Bangla. *Proceedings of ICON*, 15-24.

[7] Dasgupta, S., & Khan, D. M. (2004). Feature unification for morphological parsing in Bangla.

[8] Zhang, M., Zhang, Y., & Fu, G. (2016, August). Transition-based neural word segmentation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 421-431).

[9] Kamper, H., Jansen, A., & Goldwater, S. (2016). Unsupervised word segmentation and lexicon discovery using acoustic word embeddings. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, *24*(4), 669-679.

[10] Xie, Y. (2012). Transitional probability and word segmentation. *International Journal of English Linguistics*, *2*(6), 27.

[11] Hacker rank, Available: www.hackerearth.com

[12] Top Coder, Available: https://www.topcoder.com › community › tutorials › using-tries

[13] Stanford University, Available: http://web.stanford.edu/class/archive/cs/cs166/cs166.1146/lectures/09/Small09.pdf

[14] Princeton, Available: https://algs4.cs.princeton.edu/lectures/52Tries.pdf

[15]Towards Data Science, Available: www.towardsdatascience.com

[16] Math is Fun, Available: https://www.mathsisfun.com › data ›probability

[17] University of Texas Dallas, Available: <https://personal.utdallas.edu/~jjue/cs6352/markov/node3.html>

[18] Prothom Alo, Available: [www.prothom-alo.com](http://www.prothom-alo.com)

[19] Amader Shomoy, Available: [www.amadershomoy.com](http://www.amadershomoy.com)

[20] XlSX File, Available: [www.gmail.com/ayon.cse.neub/drive](http://www.gmail.com/ayon.cse.neub/drive)

[21] Jutivushon Chaki,2007. “বাংলা ভাষার ব্যাকরন”