B.Sc. in Computer Science and Engineering Thesis

Development of A Word Based Spell Checker for Bangla Language

Submitted by

Kowshik Bhowmik
201114033

Afsana Zarin Chowdhury
201114049

Sushmita Mondal
201114058

Supervised by

Dr. Hasan Sarwar

Professor and Head of the Department, CSE

United International University (UIU)

Department of Computer Science and Engineering
Military Institute of Science and Technology

December 2014

CERTIFICATION

This thesis paper titled **"Development of A Word Based Spell Checker for Bangla Language"**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in December 2014.

Group Members:

Kowshik Bhowmik

Afsana Zarin Chowdhury

Sushmita Mondal

Supervisor:

Dr. Hasan Sarwar
Professor and Head of the Department, CSE
United International University (UIU)

CANDIDATES' DECLARATION

This is to certify that the work presented in this thesis paper, titled, "Development of A Word Based Spell Checker for Bangla Language", is the outcome of the investigation and research carried out by the following students under the supervision of Dr. Hasan Sarwar, Professor and Head of the Department, CSE, United International University(UIU).

It is also declared that neither this thesis paper nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Kowshik Bhowmik
201114033

Afsana Zarin Chowdhury
201114049

Sushmita Mondal
201114058

ACKNOWLEDGEMENT

We are thankful to Almighty Allah for his blessings for the successful completion of our thesis. Our heartiest gratitude, profound indebtedness and deep respect go to our supervisor, Dr. Hasan Sarwar, Professor and Head of the Department, CSE, United International University(UIU), for his constant supervision, affectionate guidance and great encouragement and motivation. His keen interest on the topic and valuable advices throughout the study was of great help in completing thesis.

We are especially grateful to the Department of Computer Science and Engineering (CSE) of Military Institute of Science and Technology (MIST) for providing their all out support during the thesis work. Also, we want to express our deepest gratitude to the reviewers, Lecturer Jahidul Arafat and Lecturer Wali Mohammad Abdullah for their valuable inputs which helped us in revising the initial draft and preparing the final paper.

Finally, we would like to thank our families and our course mates for their appreciable assistance, patience and suggestions during the course of our thesis.

Dhaka
December 2014

Kowshik Bhowmik

Afsana Zarin Chowdhury

Sushmita Mondal

ABSTRACT

We present a word based Bangla spelling checker which improves the quality of suggestions with the help of the previous and next words of the misspelled words in a document. A spell checker, as we know, is a tool used for checking the spelling errors. Also, it corrects those errors in the text or a document. Development of any application for Bangla language is relatively complicated due to the complexities of the Bangla character set. Bangla alphabet consists of nearly 160 complex shaped compound character classes in addition to 50 basic character classes. Obviously, developing a spell checker application for Bangla language raises many new difficulties which do not have to be dealt with in case of a Latin based texts such as English. Most of the currently available Bangla spell checkers are based on correcting errors that has been committed on a character level. The complex rules for Bangla spelling and the complexities of Bangla character set demand different error detection & correction methods from those used for other languages. Additionally, the presence of similarly shaped characters, compound characters and the inflectional nature of the laguage present a significant challenge in producing suggestions for a misspelled word when employing the traditional methods. Considering the intricacies of the problem we have proposed, in this paper, the development of a word based spell checker for Bangla language. Our research is aimed at the correction of misspelled words by considering their neighbouring words. For this purpose we have built a lexicon of unique structure. Based on this specially built lexicon the proposed system attempts to predict the misspelled words in an input text file. This paper also shows the performance and evaluation of our proposed solution. Finally, we conclude by describing the limitations of the system with possible future improvements.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

IR : Information Retieval
IE : Information Extraction
NLG : Natural Language Generator
NLU : Natural Language Understanding
OCR : Optical Character Recognition
SR : Speech Recognition

CHAPTER 1
INTRODUCTION

1.1 Overview of Bangla spelling Checker

There are more than 200 million native speakers of Bangla, the majority of whom live in Bangladesh and in the Indian state of West Bengal. However, there has been very little research effort in the computerization of the Bangla language, leading to a dearth of Bangla natural language processing applications and tools. A Bangla spelling checker, one such application, is an essential component of many of the common desktop applications such as word processors as well as the more exotic applications, such as a machine language translator. One particular challenge facing the development of a usable spelling checker for Bangla is the languages complex orthographic rules, in part a result of the large gap between the spelling and pronunciation of a word. One impact of this complexity can be seen in the observation that two of the most common reasons for misspelling are (i) phonetic similarity of Bangla characters and (ii) the difference between grapheme representation and phonetic utterances. While there has been a sustained effort of late to develop a usable spelling checker, none of the solutions has been able to handle the full orthographic complexity of Bangla. We use the steps in the process of checking the spelling of a word when:

(a) detect whether it is misspelled or not,
(b) generate suggestions if it is misspelled, and
(c) rank the suggestions so that the most likely candidate is placed first.

1.2 What is Bangla Spelling Checker?

It is desirable from a speller that it would search through a document for invalid or miss-spelled words. The searching area might be pre-selected by highlighting the portion of the document or the checking should run forward starting from the cursor position of the active document up to the end. Functionally, each word is identified on the run and the word is matched with the database of the valid word-stock or word dictionary. If no match is found the word is an invalid or miss-spelled one. In case of Bangla the development is relatively tedious due to language complexities. Like other languages, Bangla is divided

into vowels and consonants. Half-form of vowels exist when vowels guide the sound of consonants. The place where the half-form would reside is not common, a few resides left of the biasing consonants, called as left-biased, similarly there are right-biased, bottom biased even both-biased half forms of vowels. Compound form of consonants, half-form of compound consonants including plain consonants exist in character set at the application level. Though Bangla alphabet is case-insensitive, Bangla fonts do have about 190 different characters of which 50 are in principal forms.

Human beings often produce sentences that are ill-formed at various levels, including the typographical/morphological, syntactic, semantic, and pragmatic levels. When we encounter ill-formed sentences we usually understand their meanings and tolerate the errors. The intended meaning can often be inferred by using a variety of linguistic and real world knowledge.

Shanon tested the interpretation of spoken ungrammatical sentencesin Hebrew, with one to three errors in a sentence [2]. The errors included violation of agreement rules (e.g. number,gender, and tense). Shanon found that humans preferred particular types of correction, for example in number-agreement violation, the verb is replaced rather than the noun, and in tense-agreement between a verb and an adverbial phrase, the verb is replaced rather than the adverbial phrase. He also found that unmarked forms such as verbs in infinitive form are replaced more than inflected forms. Shanon also observed a least effort principle: if there are two possible changes - for example a single change or a double change -, then the simpler change is preferred.

An experiment by Cooper and others explored humans' tolerance of ill-formed sentences [3]. Their results, based on spoken text which included missing words showed that listeners could detect errors better when they paid attention to detecting errors than when they didn't pay attention to this; listeners were highly accurate in reporting the presence of missing words (96 % of 190 cases) when they were forewarned about the specific types of errors that might be encountered while listening. But the detection of missing words (34 % of 80 cases) was quite poor when the words were easily predictable from context and when listeners were not forewarned about the specific types of errors.

1.3 Stemmer use in Bangla Spelling Checker

Stemming is a process by which a word is split into its stem and affix. Terms with common stems tend to have similar meaning, which makes stemming an attractive option to increase the performance of spelling checkers and other information retrieval applications. Another advantage of stemming is that it can drastically reduce the dictionary sized used in various NLP applications, especially for highly inflected languages. The design of stemmers is

language specific, and requires some to significant linguistic expertise in the language, as well as the understanding of the needs for a spelling checker for that language [4] . Consequently, a stemmers performance and effectiveness in applications such as spelling checker vary across languages. A typical simple stemmer algorithm involves removing suffixes using a list of frequent suffixes, while a more complex one would use morphological knowledge to derive a stem from the words. The various stemming algorithms have been evaluated in various applications from spelling checker to information retrieval, and the results show that stemming appears to be more effective in such applications for highly inflected languages.

1.4 How we use the Bangla Spelling Checker

The detection and correction of ill-formed sentences including a single syntactic error introduced by replacement of a valid word by a known/unknown word, insertion of an extra known/unknown word, or by deletion of a word. Many systems have focussed on the recovery of ill-formed texts at the typographical level (Damerau, 1964; Damerau and Mays,1989), the morpho-syntactic level (Vosse, 1992), the syntactic level (Hayes and Mouradian,1981; Mellish, 1989), the semantic level (Fass and Wilks, 1983), and the pragmatic level(Granger, 1983). Those systems described how to identify a localised error and how to repair it using grammar independent rules (Mellish, 1989), grammar specific rules (meta-rules)(Weischedel and Sondheimer, 1983), semantic preferences (Fass and Wilks, 1983), and heuristic approaches (Damerau, 1964; Damerau and Mays, 1989; Vosse, 1992) [4].

Weischedel and Sondheimer used the term one-stage error recovery for a system that can process both ill-formed and well-formed sentences with a single parser [4]. Our system is a two-stage error recovery parser based on chart parsing and consists of a well-formed sentence chart parser (WFSCP) and an ill-formed sentence chart parser (IFSCP) with a spelling correction algorithm based on dictionary lookup. The system invokes IFSCP only if the WFSCP cannot recognise the input string as well-formed. When the IFSCP identifies a local error to be substitution of a word, the spelling correction algorithm is invoked and provided with syntactic information inferred by IFSCP, to correct the word which is believed to be misspelt. This strategy has the advantage that the recovery process cannot affect the performance of a parser on well-formed sentences in terms of speed or complexity of parsing strategies.

This is an advantage in terms of processing efficiency: in terms of human processing, it corresponds to proposing that human processing has efficiency as a goal, and that consequently special methods that are not normally used are brought to bear on ill-formed sentences. Weischedel and Sondheimer would appear to be positing that error-repair methods are initiated exactly when the error is first detected - our approach corresponds to saying that it

is reasonable to defer correction until more is known. This could be the end of the current phrase, or the current sentence, or perhaps just until a couple of further words have been seen. In the WFSCP/IFSCP system, WFSCP processing of the current sentence is completed before error correction is begun, but this is largely a matter of algorithmic convenience, and the system could be adapted so that it would try correction after the end of the current noun phrase, if an error was detected while a noun phrase was being processed (for example, because an unknown word was found).

The Bangla speeling checker have also many alternatives. There are two ranking strategies: syntactic level ranking and lexical level ranking. The syntactic rank is a penalty score derived from the importance of the repaired constituent in the local tree (e.g. head constituents are more important than modifiers) and the type of error correction (e.g. substitution ¡ addition = deletion). The lexical rank depends on the distance between two letters on a keyboard. For example, held would be considered a more plausible replacement than hold for the nonword hwld, because e is closer to w on a standard keyboard than is o. Among humans, this type of correction strategy would of course only be available to those who were aware of keyboard layout. Vosse (1992) attempted to correct an ill-formed sentence at the morphosyntactic level. If there was a misspelt word, then his spelling corrector suggested the best correction. With this best correction his syntactic parser continued. Our system, on the other hand, employs a strictly top-down approach to a misspelt word. After WFSCP has finished trying to parse an ill-formed sentence containing a misspelt word, IFSCP is invoked and provided with the phrase structure, and misspelt word (if found by WFSCP). If IFSCP suggests that the error type is substitution of an unknown word, or (more difficult) a known word, then the lexical category inferred by IFSCP is used to assist in the spelling correction.

CHAPTER 2
LITERATURE REVIEW

2.1 Natural Language Processing

Natural Language Processing (NLP) is the computerized approach to analyzing text that is based on both a set of theories and a set of technologies [5]. And, being a very active area of research and development, there is not a single agreed-upon definition that would satisfy everyone, but there are some aspects, which would be part of any knowledgeable person's definition.

2.1.1 Major Tasks in NLP

The following is a list of some of the most commonly researched tasks in NLP. Note that some of these tasks have direct real-world applications, while others more commonly serve as sub-tasks that are used to aid in solving larger tasks. What distinguishes these tasks from other potential and actual NLP tasks is not only the volume of research devoted to them but the fact that for each one there is typically a well-defined problem setting, a standard metric for evaluating the task, standard corpora on which the task can be evaluated, and competitions devoted to the specific task.

Information Retrieval is concerned with storing, searching and retrieving information. It is a separate field within computer science (closer to databases), but IR relies on some NLP methods (for example, stemming) [6] . Some current research and applications seek to bridge the gap between IR and NLP. Automated information retrieval systems are used to reduce what has been called information overload. Many universities and public libraries use IR systems to provide access to books, journals and other documents.

Information extraction (IE) is the task of automatically extracting structured information from unstructured and/or semi-structured documents [7] . In most of the cases this activity concerns processing human language texts by means of Natural Language Processing (NLP). Recent activities in multimedia document processing like automatic annotation and content extraction out of images / audio / video could be seen as information extraction.

Natural Language Generation (NLG) is the natural language processing task of generat-

ing natural language from a machine representation system such as a knowledge base or a logical form [8] . Psycholinguists prefer the term language production when such formal representations are interpreted as models for mental representations. It could be said an NLG system is like a translator that converts a computer based representation into a natural language representation. However, the methods to produce the final language are different from those of a compiler due to the inherent expressibility of natural languages.

Natural Language Understanding converts chunks of text into more formal representations such as first-order logic structures that are easier for computer programs to manipulate. Natural language understanding involves the identification of the intended semantic from the multiple possible semantics which can be derived from a natural language expression which usually takes the form of organized notations of natural languages concepts. Introduction and creation of language met model and ontology are efficient however empirical solutions. An explicit formalization of natural languages semantics without confusions with implicit assumptions such as closed world assumption (CWA) vs. open world assumption, or subjective Yes/No vs. objective True/False is expected for the construction of a basis of semantics formalization.

Speech recognition (SR) is the translation of spoken words into text. It is also known as 'automatic speech recognition' (ASR), "computer speech recognition", or just 'speech to text' (STT). Some SR systems use 'speaker-independent speech recognition' while others use "training" where an individual speaker reads sections of text into the SR system. These systems analyze the person's specific voice and use it to fine-tune the recognition of that person's speech, resulting in more accurate transcription. Systems that do not use training are called 'speaker-independent' systems. Systems that use training are called 'speaker-dependent' systems [9] .

Optical Character Recognition (OCR) is generally an 'off-line' process, which analyzes a static document. Handwriting movement analysis can be used as input to handwriting recognition [10] . Instead of merely using the shapes of glyphs and words, this technique is able to capture motions, such as the order in which segments are drawn, the direction, and the pattern of putting the pen down and lifting it. This additional information can make the end-to-end process more accurate. This technology is also known as "on-line character recognition", "dynamic character recognition", "real-time character recognition", and "intelligent character recognition".

2.1.2 Levels of Natural Language Processing

The most explanatory method for presenting what actually happens within a Natural Language Processing system is by means of the 'levels of language' approach. This is also

referred to as the synchronous model of language and is distinguished from the earlier sequential model, which hypothesizes that the levels of human language processing follow one another in a strictly sequential manner. Psycholinguistic research suggests that language processing is much more dynamic, as the levels can interact in a variety of orders. Introspection reveals that we frequently use information we gain from what is typically thought of as a higher level of processing to assist in a lower level of analysis. For example, the pragmatic knowledge that the document you are reading is about biology will be used when a particular word that has several possible senses (or meanings) is encountered, and the word will be interpreted as having the biology sense.

Of necessity, the following description of levels will be presented sequentially. The key point here is that meaning is conveyed by each and every level of language and that since humans have been shown to use all levels of language to gain understanding, the more capable an NLP system is, the more levels of language it will utilize.

Phonology This level deals with the interpretation of speech sounds within and across words. There are, in fact, three types of rules used in phonological analysis: 1) phonetic rules for sounds within words; 2) phonemic rules for variations of pronunciation when words are spoken together, and; 3) prosodic rules for fluctuation in stress and intonation across a sentence. In an NLP system that accepts spoken input, the sound waves are analyzed and encoded into a digitized signal for interpretation by various rules or by comparison to the particular language model being utilized.

Morphology This level deals with the nature of components of words, which are composed of morphemes the smallest units of meaning. For example, the word preregistration can be morphologically analyzed into three separate morphemes: the prefix pre, the root registration, and the suffix. Since the meaning of each morpheme remains the same across words, humans can break down an unknown word into its constituent morphemes in order to understand its meaning. Similarly, an NLP system can recognize the meaning conveyed by each morpheme in order to gain and represent meaning. For example, adding the suffixed to a verb, conveys that the action of the verb took place in the past. This is a key piece of meaning, and in fact, is frequently only evidenced in a text by the use of the morpheme.

Lexical At this level, humans, as well as NLP systems, interpret the meaning of individual words. Several types of processing contribute to word-level understanding the first of these being assignment of a single part-of-speech tag to each word. In this processing, words that can function as more than one part-of-speech are assigned the most probable parts of speech tag based on the context in which they occur. Additionally at the lexical level, those words that have only one possible sense or meaning can be replaced by a semantic representation of that meaning. The nature of the representation varies according to the semantic theory utilized in the NLP system .

The lexical level may require a lexicon, and the particular approach taken by an NLP system will determine whether a lexicon will be utilized, as well as the nature and extent of information that is encoded in the lexicon. Lexicons may be quite simple, with only the words and their part(s)-of-speech, or may be increasingly complex and contain information on the semantic class of the word, what arguments it takes, and the semantic limitations on these arguments, definitions of the sense(s) in the semantic representation utilized in the particular system, and even the semantic field in which each sense of a polysemous word is used.

Syntactic This level focuses on analyzing the words in a sentence so as to uncover the grammatical structure of the sentence. This requires both a grammar and a parser. The output of this level of processing is a (possibly de-linearized) representation of the sentence that reveals the structural dependency relationships between the words. There are various grammars that can be utilized, and which will, in turn, impact the choice of a parser. Not all NLP applications require a full parse of sentences, therefore the remaining challenges in parsing of prepositional phrase attachment and conjunction scoping no longer stymie those applications for which phrasal and clausal dependencies are sufficient. Syntax conveys meaning in most languages because order and dependency contribute to meaning. For example the two sentences: 'The dog chased the cat.' and 'The cat chased the dog.' differ only in terms of syntax, yet convey quite different meanings.

Semantics This is the level at which most people think meaning is determined, however, as we can see in the above defining of the levels, it is all the levels that contribute to meaning. Semantic processing determines the possible meanings of a sentence by focusing on the interactions among word-level meanings in the sentence. This level of processing can include the semantic disambiguation of words with multiple senses; in an analogous way to how syntactic disambiguation of words that can function as multiple parts-of-speech is accomplished at the syntactic level. Semantic disambiguation permits one and only one sense of polysemous words to be selected and included in the semantic representation of the sentence. For example, amongst other meanings, 'file' as a noun can mean either a folder for storing papers, or a tool to shape one's fingernails, or a line of individuals in a queue. If information from the rest of the sentence were required for the disambiguation, the semantic, not the lexical level, would do the disambiguation. A wide range of methods can be implemented to accomplish the disambiguation, some which require information as to the frequency with which each sense occurs in a particular corpus of interest, or in general usage, some which require consideration of the local context, and others which utilize pragmatic knowledge of the domain of the document.

Discourse While syntax and semantics work with sentence-length units, the discourse level of NLP works with units of text longer than a sentence. That is, it does not interpret multi sentence texts as just concatenated sentences, each of which can be interpreted singly. Rather, discourse focuses on the properties of the text as a whole that convey meaning by

making connections between component sentences. Several types of discourse processing can occur at this level, two of the most common being anaphora resolution and discourse/-text structure recognition. Discourse/text structure recognition determines the functions of sentences in the text, which, in turn, adds to the meaningful representation of the text.

Pragmatic This level is concerned with the purposeful use of language in situations and utilizes context over and above the contents of the text for understanding The goal is to explain how extra meaning is read into texts without actually being encoded in them. This requires much world knowledge, including the understanding of intentions, plans, and goals. Some NLP applications may utilize knowledge bases and inference modules.

2.2 Noisy Optical Character Recognition and Bangla Language

For some decades there has been massive, expensive, ongoing institutional digitization of textual resources such as books, magazines, newspaper articles, documents, pamphlets and ephemera from cultural archives. In addition, declassified government documents are being released into the public domain, and many organizations and individuals are converting existing document images into machine readable text via OCR [11] . To make this digitization accurate and efficient, a great deal of research work has been done for languages based on the Latin script.

Although Bangla is one of the most widely spoken languages (over 200 million people use Bangla as their medium of Communication) of the world, research is acute in recognition of Bangla characters. Under this context, an effort has been taken globally to computerize the Bangla language. Compared to English and other language scripts, one of the major stumbling blocks in Optical Character Recognition (OCR) of Bangla script is the large number of complex shaped character classes of Bangla alphabet. In addition to 50 basic character classes, there are nearly 160 complex shaped compound character classes in Bangla alphabet. Dealing with such a large variety of characters with a suitably designed feature set is a challenging problem. Uncertainty and imprecision is inherent in handwritten script. Moreover, such a large variety of complex shaped characters, some of which have close resemblance, makes the problem of OCR of Bangla characters more difficult.

2.3 Degraded Text

The biggest problem associated with the retrieval of OCR text from scanned data is the unavoidable character corruption that results even from the best of the OCR systems. Very few researches have dealt with this problem. In case of good quality input, OCR errors have little to no effect at all. But the effectiveness is reduced to great extent in case of poor

scanning quality. But it is not always possible to supply good quality scans as input to the OCR system [12] . As OCR often deals with the digitization process of old documents or literature, scan quality, font type and font size of the input will not always be compatible with that of any standard OCR system. As a result, recognition errors can occur during the preprocessing stages of the OCR system. Due to recognition errors in the character recognition process, unpredictable distortion occurs which is a major problem with retrieval of OCR.Users have no idea of distortion of such sorts. As a consequence, their query hardly match the terms that are actually stored in the OCR text. This the efficiency of retreival is greatly reduced. This case is observed more often in case of low quality inputs. Fault tolerant retrieval strategy which is based on automatic keyword extraction & fuzzy matching is proposed with a view to reducing the losses derived from retrieving noisy OCR text.

2.3.1 Types of Spelling Error

Techniques were designed on the basis of spelling errors trends these are also called error patterns. Therefore many studies were performed to analyze the types and the trends of spelling errors. The most notable among these are the studies performed by Damerau. According to these studies spelling errors are generally divided into two types Typographic errors and Cognitive errors [13] .

A. Typographic errors These errors are occurring when the correct spelling of the word is known but the word is mistyped by mistake. These errors are mostly related to the keyboard and therefore do not follow any linguistic criteria. A study by Damerau shows that 80% of the typographic errors fall into one of the following four categories.

1. Single letter insertion; e.g. typing acress for cress.
2. Single letter deletion, e.g. typing acress for ctress.
3. Single letter substitution, e.g. typing acress for across.
4. Transposition of two adjacent letters, e.g. typing acress for caress.

The errors produced by any one of the above editing operations are also called single-errors.

The rate of errors for Bangla is shown in [14] Table 2.1.

The number of characters making the error is shown in Table 2.2

B. Cognitive errors These are errors occurring when the correct spellings of the word are not known. In the case of cognitive errors, the pronunciation of misspelled word is the same or similar to the pronunciation of the intended correct word. (e.g. receive -> recieve, abyss -> abiss).

Table 2.1: Rate of Error for Bangla

Types of Error	Percentage
Substitution / Replacement	66.32
Deletion Error	21.88
Insertion Error	6.53
Swap / Transposition Error	5.27

Table 2.2: Number of Characters Making Error

Error Zone Length (in no. of characters)	% of words
1	41.36
2	32.94
3	16.58
4	7.1
5	1.78
6	0.24

2.4 Post-Processing

For a complex language like Bengali even the most complex pre-processing and character recognition algorithms combined together is not sometimes enough to produce a satisfactory result. The use of compound words in the language poses a major problem for the recognition process. Also most application of Bengali OCR deals with the digitization of old documents whose font type and sizes vary drastically.These factors greatly affects the accuracy and the efficiency of Bengali OCR systems [15] .

But the principal concern of the current OCR system is to achieve maximum accuracy. To attain the desired level of accuracy a post-processing recognition engine is added to the system. This post-processing recognition engine consists of filters such as a font engine, a lexicon filter, a near neighbor analysis filter. The post-processing layer is the integration of different analyzers as shown in Figure 2.1. By combining the post-processing layer or in other words, the font engine, lexicon filter, near neighbor analyzer with further analyzing filters such as pattern matching filter, grammar checker and the knowledge driven analyzer forms an efficient algorithm that ensures a much higher accuracy of the output.

2.5 Font Engine

After preprocessing the digitized image the data is sent for matching to the font engine filter. The filter then maps its input to its database. A probabilistic model is used for the

comparison between the input data and the database of the fonts.The engine matches each of the fonts derived from the digitized image to the font database that is contained in the engine. Then, using the probabilistic model, a threshold value is calculated [15] . The image that gives the highest threshold value is selected. Then the selected value is mapped

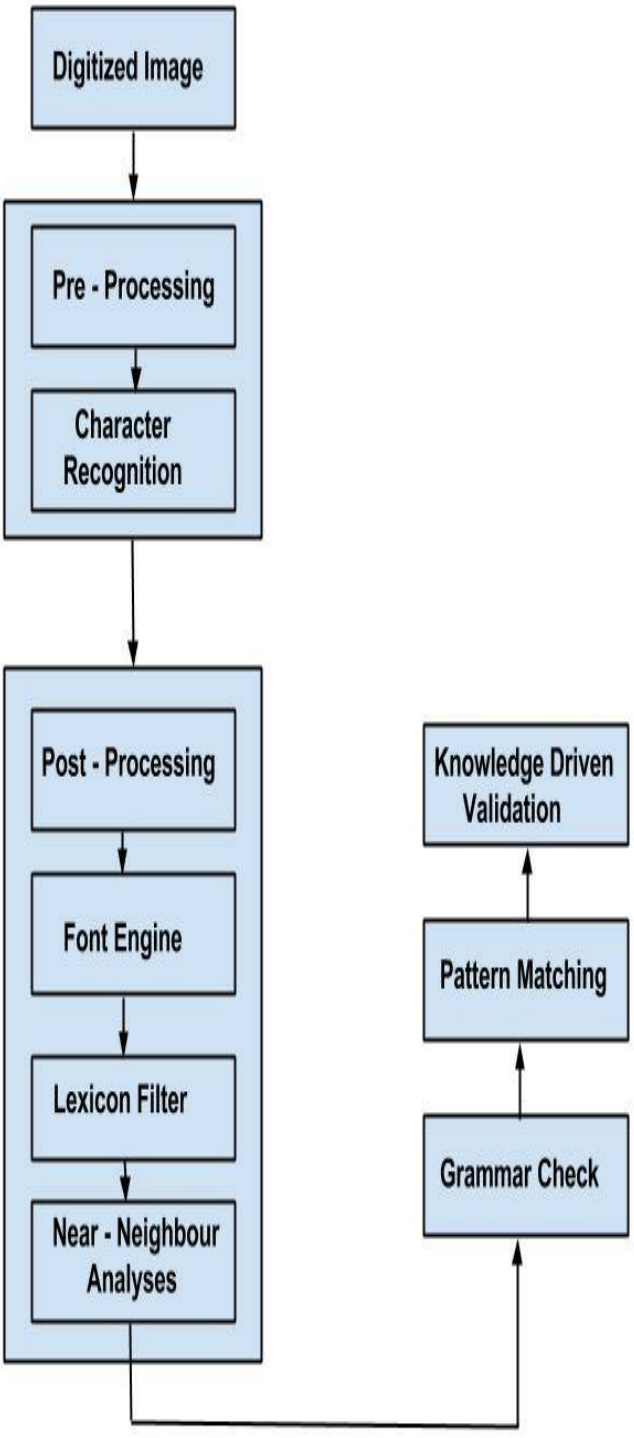


Figure 2.1: Block Diagram of Present System

to the font. Brute force algorithm is used to do the mapping. Finally, for retrieval purposes in the future, the successfully analyzed and mapped fonts are stored to the database.

2.6 Lexicon Filter

The accuracy of the out of the OCR system can be increased by constraining the out using a lexicon. Lexicon, as we know, is a list of allowed words in a particular document. For example all the words in the Bengali language is allowed to occur in a document. But in a more practical OCR system which is designed for a specific field. In that case, the size of the lexicon naturally narrows down. This method poses some problems if the word in the document is not in the lexicon of that system. In that case, periodic updating of the lexicon is required [15].

Practical documents are more likely to contain words or strings made up of characters rather than characters that are isolated [15] . The efficiency of a lexicon-driven string recognition can be improved if we match the string classes rather than matching them one by one.In case of string classes matching, the strings with low score partial match are discarded. The search strategy is designed in such a way that the lexicon is organized in the form of a tree or a graph. This ensures that common substrings and the string image are matched only once.

2.6.1 Structure of a Lexicon

Trees can be use as the implementation of a lexicon data type. This tree stores a large collection of words and makes entry and lookup fast. The structure developed by Edward Fredkin in 1960, is called a trie taken from the central letters of 'retieval'. The trie based lexicon structure makes it possible to determine whether a word is in the dictionary more quickly than you can using a hash table, and it offer natural support for confirming the presence of prefixes in a way that hash tables cant. On one level, a trie is simply a tree in which each node branches in as many as 256 different directions, one for each position in the ASCII table.

Digital search trees store strings character by character.Figure 2.2 is a tree that represents the same set of 12 words; each input word is shown beneath the node that represents it. (Two-letter words lead to prettier pictures; all structures that we'll see can store variable-length words.) In a tree representing words of lowercase letters, each node has 26-way branching (though most branches are empty, and not shown in Figure 2.2). Searches are very fast: A search for "is" starts at the root, takes the "i" branch, then the "s" branch, and ends at the desired node. At every node, we access an array element (one of 26), test for null, and take a

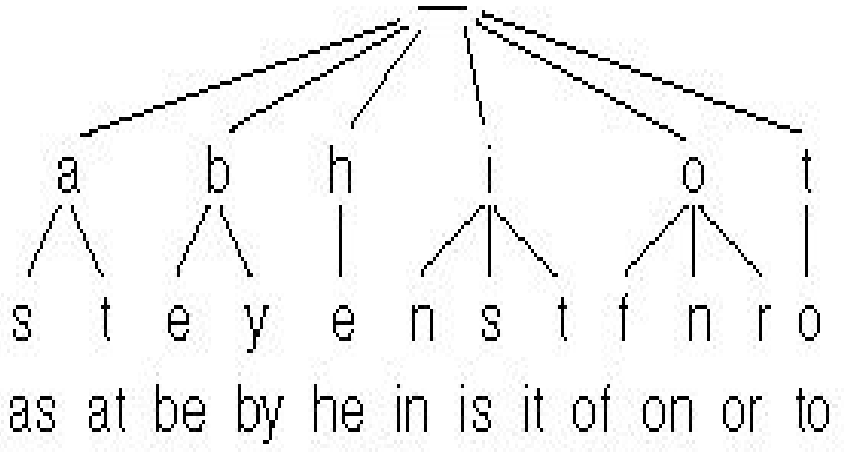


Figure 2.2: Digital Search Tree

branch. Unfortunately, search tries have exorbitant space requirements: Nodes with 26-way branching typically occupy 104 bytes, and 256-way nodes consume a kilobyte. Eight nodes representing the 34,000-character Unicode Standard would together require more than a megabyte! [16]

When using a tree to back a lexicon, the words are implicitly represented by the tree itself, each word represented as a chain of links moving downward from the root [15] . The root of the tree corresponds to the empty string, and each successive level of the tree corresponds to the prefix of the entire word list formed by appending another letter to the string represented by its Parent. The A link descending from the root leads to the sub-tree containing all of the words beginning with A, the B link from that node leads to the sub-tree containing all of the words beginning with AB, etc. Each node stores a true whenever the substring that ends at that particular point is a legitimate word. If we pretend that the English alphabet only has 7 letters (let's say A through G) and we further assume that the English language only has five wordsbe, bed, cab, cage, and cagedthen the underlying tree structure of the English Lexicon would look like figure 2.something.

2.6.2 Two Tree Implementation of Bangla Lexicon

There are two lexicons in the system under discussion : one for the root words and the other for the suffixes.

The root word lexicon is organized in alphabetical order as follows. It contains all the root words as well as morphologically deformed variants of the root words so that simple concatenation with suffixes produces valid words. Some of the morphologically deformed

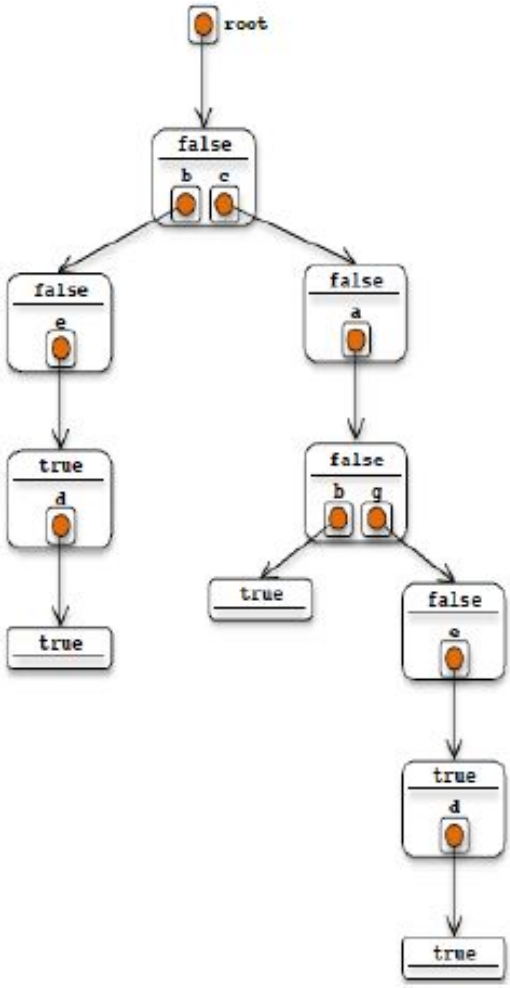


Figure 2.3: Tree Structure in English Lexicon

variants cannot be used in isolation as root words in the text. This information along with parts of speech and other grammatical information are tagged with each entry of the root lexicon in the form of a number.Root words of length up to 3 characters as well as the distinct strings of root words of length greater than 3 are represented in a tree structure. At each leaf node of the trie an address as well as a boolean flag is maintained. The address of a leaf node Lk (where Lk denotes the kth node of Lth level) of the trie points to the first word of the main lexicon, the first three characters of which are the same as those we encounter by traversing the tree starting from the root word and continuing up to that leaf node Lk. Therefore, if a string is of length four or more (characters), then we can obtain the address from the trie by using the first three characters and then sequentially searching the words in main lexicon.The lexicon structure of error correction is shown in Figure 2.4. The boolean flag of a leaf node Lk indicates whether the string of the characters obtained by traversing the tree, starting from the root node and continuing up to Lk, is a valid root word or not. Thus, by means of boolean flags, we can detect the validity of test strings whose lengths are

less than or equal to 3, without searching in the root word lexicon.

The suffix lexicon is maintained in another tree structure. Each node in the tree represents either a valid suffix or an intermediate character of a valid suffix. A node representing a valid suffix is associated with a marker needed for a grammatical agreement test [17] .If one traverses from the root of the suffix lexicon trie to a node having a valid suffix marker, then the encountered character sequence represents a valid suffix string in reverse order (i.e., last character first). This is so because during suffix search, we start from the last character of a string and move leftwards.

2.6.3 Generation of Suspicious Words

In the OCR validation module used by the System for Preservation of Electronic Recourses (SPER) which was developed at the U.S. National Library of Medicine, the suspicious words in an OCR output of scanned document are detected and corrected. To make corrections of the suspicious words, first it derives the partial format of each suspicious word. Then the module retrieves candidate words by searching the lexicon using partial-match search. Then the suspicious words and the words derived by the search are compared taking into consideration the joint probabilities of N-gram and OCR edit.transformation corresponding to the candidates. The derivation of partial format which is based on the error analysis of the OCR engine effectively generates the desired candidate words from the lexicon. This lexicon is represented by ternary search trees.

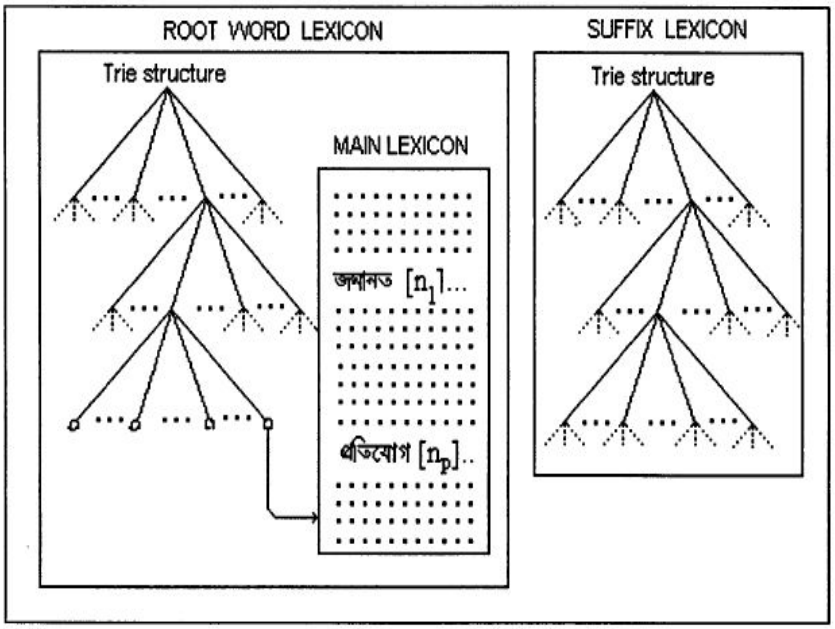


Figure 2.4: Lexicon structure for error correction

Table 2.3: Assumptions of Conversion

	Character Conversions	Examples
1	ce*	CHARGECHABGE
2	ce1*e2	informationinforrnation
3	ce1e2*	AdulterationAclulteration
4	ce1*e2*	SulphateSiilphate
5	c1c2e*	libellantlibeUant

The FineReader OCR Engine used by SPER assigns a confidence value to each recognized character by a binary feature isSuspicious. In addition, OCR words that are not found in the built-in dictionaries may also be indicated by a binary feature isFromDict. So using this OCR engine we have two types of words that are considered to be suspicious. The first kind is the words that contains one or several characters whose confidence value is below the threshold value allowed by that engine [18] . The second type of suspicious words include the words that are not found in context-based lexicon of the scanned documents.

2.6.4 Word Partial Format Derivation

A partial word is a word that contains one or several wild cards . where a wild card signifies a character which is unrecognized. Word partial format derivation is the process of estimating the partial format of a word that is produced as an output of the OCR engine by combining the position of the characters with low confidence value within the word. Then we use this word partial format to generate candidate words from the lexicon by approximate string matching.

Errors in a word produced by the OCR engine can be considered as transformation errors of one or more characters. In this particular application three kinds of transformation errors are taken into account; shown in Table 2.3. Firstly, there is the error where a character is transformed into another character. Secondly, error occurs then a single character is transformed into two characters. Finally, there is the case where two characters are transformed into one character. Considering the different positions of a character that has gone through a transformation error, or in other word the character that has lower confidence value than the defined threshold, we have five kinds of transformation errors. These five errors are signified in Table 2.3 where 'c' denotes a character which is correct and 'e' denotes the error character. Low confidence characters are denoted by '*' .

The process of deriving partial format is aimed at estimating the partial formats by sequentially replacing characters with low confidence value and their neighbors with proper number of wild cards. Different replacements results in different transformation rules. So a suspicious word with t low confidence characters could generate up to 4t partial formats. The

maximum value occurs when every one of the low confidence characters appear inside the word. Also it has to be taken into consideration that none of the low confidence character inside the word is either next to another low confidence character or share the same neighbor with another low confidence character.

2.6.5 Approximate String Matching Methods

Cross correlation matching method The cross-correlation function is commonly used in image and signal processing where an unknown signal is searched for a known feature or shape, and is sometimes described as a sliding dot-product. In this project, the cross-correlation function was modified to operate on words and letters instead of quantitative signals. The function compares words a (of length m) and b (of length n ¿= m) and produces a vector W of length l = m + n - 1 with elements Wi (where i = 0 ... l - 1) defined by the equation below:

$$W_i(a,b) = \begin{cases} \sum_{j=0}^{i} S(a_j, b_{(n-1)-i+j}) & \text{if } i = 0...m-1 \\ \sum_{j=0}^{m-1} S(a_j, b_{(n-1)-i+j}) & \text{if } i = m...n-1, m \neq n \\ \sum_{j=0}^{(l-1)-i} S(a_{(n-1)-(l-1-i)+j}, b_j) & \text{if } i = n...l-1 \end{cases}$$

Figure 2.5: Cross correlation equation-1

where

$$S(x,y) = \begin{cases} 1 \text{ if } x = y \\ 0 \text{ if } x \neq y \end{cases}$$

Figure 2.6: Cross correlation equation-2

This function compares the two words by aligning them against each other and examining the corresponding pairs of letters. The element W0 is defined as the alignment position where the last letter of word b aligns with the first letter of word a. Example calculations using the true word Biology and the mistranslated OCR word 1Biology are shown in Figure 2.7 below:

Vector Element	W_0(Biology, 1Biologv)	W_4(Biology, 1Biologv)	W_6(Biology, 1Biologv)
Word alignment	Biology 1Biologv 0	Biology 1Biologv 00100	Biology 1Biologv 1111110
Score	0	0+0+1+0+0=1	1+1+1+1+1+1+0=6

Figure 2.7: Misunderstand OCR word

Approximate String Matching by Ternary Search Tree

The Context based lexicon that is used by the OCR engines is represented by Ternary search trees [18] . The ternary search trees are more efficient than hashing and other search methods as they offer efficient algorithms for sorting and searching strings.Each node of the Ternary Search Tree contains a split character and three pointers to its low, high and equal child nodes (based on the split character). During the sorting or searching of a word, the characters in the query word are compared one by one with the split characters of nodes in the search path. This helps in deciding whether the next direction of search among the possible three. This helps find the optimum search path. New nodes are entered at the finishing of the search path when sorting operations are carried out. In the case of searching: a boolean value true will be returned when the search process ends in comparing the last character of the word and the split character of a leaf node, and finds them be equal, otherwise false will be returned. Adjusting the splitting rules can help achieve more sophisticated methods of searching such as partial-match search.

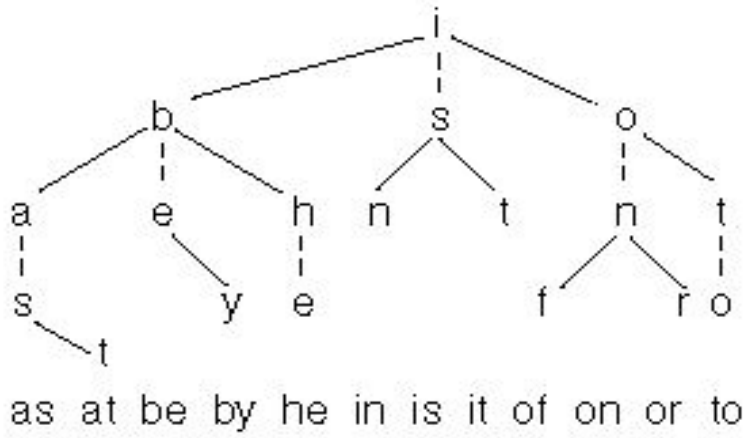


Figure 2.8: Probability of match method

Figure 2.8 something is a balanced ternary search tree for the same set of 12 words. The low and high pointers are shown as solid lines, while equal pointers are shown as dashed lines. Each input word is shown beneath its terminal node [16] . A search for the word ”is” starts at the root, proceeds down the equal child to the node with value ”s,” and stops there after two comparisons. A search for ”ax” makes three comparisons to the first letter (”a”) and two comparisons to the second letter (”x”) before reporting that the word is not in the tree.

Probability of match method The probability of match method is the edit distance method with the addition of the probabilistic substitution matrix. The edit distance algorithm was implemented slightly differently with this method in order to directly calculate the probability that a true word a would be mistranslated into the OCR word b. In order to calculate this result, the probability of the OCR making each edit was looked up in the substitution matrix, and all of the letter-wise probabilities were multiplied together to form the word translation probability. This implementation also required modifying the algorithm to find the maximum probability of translation rather than the minimum number of edits. The recurrence relation became, then

$$D(a_i, b_j) = \max \begin{pmatrix} D(a_i, b_{j-1}) \cdot P_I(b_i) \\ D(a_{i-1}, b_{j-1}) \cdot P_M(a_i, b_i) \\ D(a_{i-1}, b_j) \cdot P_D(a_i) \end{pmatrix}$$

Figure 2.9: Recurrence relation

where PI(x)andPD(x)are the frequency of insertion and deletion of the character x, and PM(x,y) the frequency of substituting x for y. [approximate]

Similarity Keys A key is assigned to each dictionary word an only dictionary keys are compared with the key computed for the non word [13] .The word for which the keys computed for the non word .The word for which the keys are most similar are selected as suggestion .such an approach is speed effective as only the words with similar keys have to be processed With a good transformation algorithm this method can handle keyboard errors.

N-gram Model Formulation

In this formulation technique, N-grams are n-letter sub sequences of words or strings. Here n usually is one, two or three. One letter ngrams are referred to as unigrams or monograms;

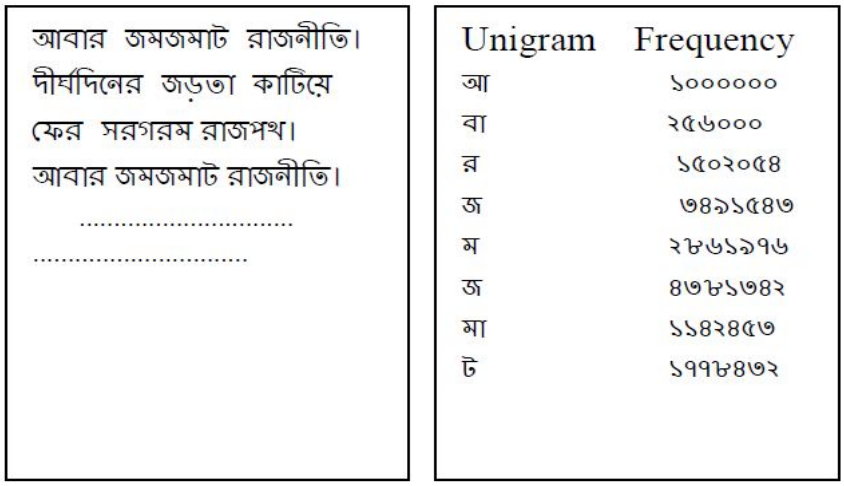


Figure 2.10: Training N-gram Model(a) [1]

two letter n-grams are referred to as bi-grams and three letter n-grams as trigrams. In general, n-gram detection technique work by examining each n-gram is an input string and looking it up in a pre-compiled table of n-gram statistics to as certain either its existence or its frequency of words or strings that are found to contain nonexistence or highly infrequent n-grams are identified as either misspellings.

Letter n-grams, including tri-grams, bi-grams and unigrams have been used in a variety of ways in text recognition and spelling correction techniques. They have been used by OCR correctors to capture the lexical syntax of a dictionary and to suggest legal corrections [19] .

Edit Distance Matching Method The edit distance algorithm is a well-known recursive algorithm which supplies the optimum alignment between two ordered strings and determines the global minimum number of edits needed to transform one string into the other. In this algorithm the edits can be insertions, deletions, or substitutions. The algorithm is often implemented using a dynamic programming approach that calculates the number of edits D between every possible left-sided substring of each of the two words a and b. D(ai,bj), for example, is the edit distance between the first i letters of the word a and the first j letters of the word b. The dynamic programming calculation is recursive, with

where CI, CM, and CD, are the costs of insertion, match/substitution, and deletion respectively. In the simplest case, the costs of insertion, deletion, and substitution are all unit costs, and the cost of a match is zero. That is, CI(x) = CD(x) = 1 for all x, and CM(x,y) = 0 if

x = y and 1 otherwise. If the two words a and b have lengths m and n, then D(am,bn) will contain the minimum edit distance needed to transform a into b. In our use of this function, normalizing by word length was desirable, so the final matching score was determined by D/m, giving the edit distance as a percentage of the true word length.

The method can be extended to use different costs for different edits, so that CM(x,y) may depend on the frequency of the OCR substitutions of x for y, CI(x) on the frequency of inserting character x, and CD(x) on the frequency of deleting character y, in a similar way to that described for the cross correlation method above. This is commonly implemented using a substitution matrix as described above. For this study, the edit distance algorithm was implemented first without and then with a substitution matrix. The implementation with a substitution matrix is described in the next section.

Bayesian probability matching method The Bayesian probability matching method further extends the edit distance algorithm, and considers the probability of match with the

Bigram	Probability
আ বা	০.৯৮
বা র	০.৭৮
জ ম	০.৭২
ম জ	০.৬৭
জ মা	০.৮৮
মা ট	০.৭৯

Trigram	Probability
আ বা র	০.৯৩
জ ম জ	০.৮২
ম জ মা	০.৭৯
জ মা ট	০.৯৮

Quadrigram	Probability
জ ম জ মা	০.৭১
ম জ মা ট	০.৬৬

Figure 2.11: Training N-gram Model (b) [1]

$$D(a_i, b_j) = \min \begin{pmatrix} D(a_i, b_{j-1}) + C_I(b_i) \\ D(a_{i-1}, b_{j-1}) + C_M(a_i, b_i) \\ D(a_{i-1}, b_j) + C_D(a_i) \end{pmatrix}$$

Figure 2.12: Minimum Edit Distance Equation

$$\mathrm{P}(t|o) = \frac{\mathrm{P}(o|t)\mathrm{P}(t)}{\mathrm{P}(o)}$$

Figure 2.13: Bayes Probability Matching Equation-1

dictionary word (using the above probability function), the frequency of occurrence of the dictionary word and the probabilities and frequencies of all the other dictionary words when calculating the ranking score. The basic Bayes equation is:

Where: P(t—o) is the probability that the true word is t given that the OCR word is o. P(o—t) is the probability that the OCR will output the word o when presented with the dictionary word t. (This is the probability of match function above.) P(t) is the frequency of the word t in the dictionary, and P(o) is the frequency of the word o in the OCR output space.

It may at first seem problematic to find P(o), which would need a huge database of OCR output to directly count the frequencies of each unique mistranslation. Fortunately, the quantity can be calculated by summing the probability of producing o from each true word in the dictionary, weighted by the probability of that true word occurring in the affiliations text. In mathematical terms,

$$\mathrm{P}(o) = \sum_i \mathrm{P}(o|t_i)\mathrm{P}(t_i)$$

Figure 2.14: Bayes Probability Matching Equation-2

where the ti are the words in the dictionary, and the sum is over all entries. It turns out that this is not even computationally expensive: P(o—ti) P(ti) is already being calculated for every [o, ti] pair since it is the numerator in the above equation, and it is only a matter of keeping a running sum as the function loops through all ti in the dictionary.

Rule-based Techniques

Rule-based methods are interesting. They work by having a set of rules that capture common spelling and typographic errors and applying these rules to the misspelled word [13] . Intuitively these rules are the inverses of common errors. Each correct word generated by this process is taken as a correction suggestion. The rules also have probabilities, making it possible to rank the suggestions by accumulating the probabilities for the applied rules. Edit distance can be viewed as a special case of a rule-based method with limitation on the possible rules.

2.7 Near-Neighbor Analysis

After being processed by the lexicon filter the data then is passed through a filter called the near-neighbor analyzer. Spell-checking is suggested using this filter [15] . After scanning the text and extracting the words of the document, the words are compared with a list containing correctly spelled words-dictionary. In concept, this method sounds very similar to that of lexicon-filtering. In actuality, it is based on language specific algorithm that handles morphology. In a highly inflated language like Bengali, spell-checker must consider a word in its various forms. For example, a word can be in its singular or plural form or in different verbal forms. These different forms are originated from that specific word and clearly these forms are inter-related. Morphology is the branch of linguistics that deals with this relation and can play quite an important role in the OCR system as different forms of the same word can occur in a document. To identify the relationship of a word in its original form and its different forms can help us a long way in developing efficient processing algorithms. So the near-neighbor analyzer is an integral part of the post-processing engine of the OCR system.

2.7.1 Rogets Thesaurus

The third edition electronic version of Rogets Thesaurus is composed of 990 sequentially numbered and named categories. There is a hierarchical structure both above and below this category level. There are two structure levels above the category level and under each of the 990 categories there are groups of words that are associated with the category heading given. The words under the categories are grouped under five possible grammatical classifications: noun, verb, adjective, adverb and preposition. These classifications are further subdivided into more closely related groups of words. Some groups of words have cross-references associated with them that point to other closely related groups of words. Figure 1 gives an example of an extract within category 373 and the grammatical classification of noun, in the thesaurus. The cross-references are given by a numerical reference to the category number

followed by the title given in brackets.

The thesaurus contains a collection of words that are grouped by their relation in meaning. Those words grouped together have a semantic relationship with each other and this information could be used to identify semantic relations between words. For example, a semantic relationship between two words could be assumed if they occurred within the same category in the thesaurus.

2.7.2 Thesaural Connections

The application of the thesaurus for the identification of semantic relations between words required a means of determining what constitutes a valid semantic connection in the thesaurus between two words. For example, given words w1 and w2. Now the question is how could the lexical organization of the thesaurus be exploited to establish whether a semantic relation w1,w2 exists between themor not. Morris and Hirst identified five types of thesaural relations between words based on the index entries of Rogets Thesaurus. For this approach four types of possible connections between words in the thesaurus were identified for the representation of semantic relations between words by considering the actual thesaural entries. This ensured the inclusion of all words located in the thesaurus, for example, those words that form part of a multi-word thesaurus entry may not be represented in an index entry. The connections that have been identified are considered between pairs of words and are outlined as follows:

(1) Same category connection is defined as a pair of words both occurring under the same category.The words would be considered to be semantically related because they were found within the same category, where a category contains a group of associated words. This connection represents the strongest connection type of the four presented because the occurrence of words within the same category indicates they are highly related and therefore have been grouped within the same area of the thesaurus.

(2) Category to cross-reference connection occurs when a word has an associated cross reference that points to the category number of another word. Cross-references occur at the end of semi-colon groups and point to other categories that closely relate to the current group of words. Therefore, the words contained under the group of words a cross-reference is pointing to are related to the current group of words that cross-reference is associated with.

(3) Cross-reference to category connection can be described as the inverse of the previous connection type given in (2). The cross-references associated with a word could be matched with the categories another word occurs under.

(4) Same cross-reference connection is defined as the cross-references of two words point-

ing to the same category number. The association of a cross-reference with a group of words indicates that the category the crossreference is pointing to contains words that are related to the current group [20] . Therefore, if two groups of words both have the same cross-references associated with them this implies that the words within these two groups could also be related.

2.7.3 Error Correction of Bangla OCR Using Morphological Parsing

Because of the inflectional structure of surface words in the Bangla language, and because the errors in our OCR output are mostly single characters per word, the effort is limited to correcting a single character in a word. Single characters also include compound characters that are grapheme combinations of two or more basic characters.

Let C be a character in the alphabet. A test is conducted a priori on a large set of data for performance evaluation of the OCR, and we let D(C) denote the subset of characters which may be wrongly recognized as C. Members of D(C) may be called confusing characters (similarly shaped characters) for C. See Figure , where misclassification percentages of characters with respect to their confusing characters are shown. Now, if a recognized string W contains a character C, and if it is a mis-recognized character, then the correct word for W is one of the strings obtained when C is replaced by elements of D(C). Strings obtained by replacing the character C with the elements of D(C) may be called alternative or candidate strings. We can attach a probability to each candidate string based on the a priori knowledge of frequency of a confusing character.

As it is not known which character in W is mis-recognized, candidate strings are generated by replacing each character in W with its confusing characters. In generating the candidates, special care should be taken about run on, split and deletion errors also. Using this technique, a complete set of candidates is generated. The candidate set thus formed is matched in decreasing order of probability against either the root word lexicon or suffix lexicon. The grammatical agreement of the candidate words is also tested for validity. The valid words are called suggested words. Among these suggested words, the first one is accepted by the system while other suggested words can be used by the user. If there is no match, the test string is rejected.

Input Char.	Misrecog-nized as	Percentage of Misclassification	Input Char.	Misrecog--nized as	Percentage of Misclassification
প	গ	2.13	ক	ফ	2.91
গ	প	3.47	ফ	ক	5.22
ল	ন	2.66	ত	ভ	1.22
ন	ল	1.97	ভ	ত	3.76
ঘ	য	8.21	য়	য	3.16
য	ঘ	4.11	য	য়	2.94
ষ	য	5.22	খ	থ	7.39
য	ষ	7.13	থ	খ	2.79
ঊ	ঊ	1.33	র	ব	2.61
ঊ	ঊ	3.21	ব	র	3.17
ঋ	ধ	1.14	্	া	3.41
ধ	ঋ	0.97	া	্	2.83
ড	ড়	2.64	এ	ঐ	1.89
ড়	ড	3.79	ঐ	এ	2.32
ু	ৃ	1.72	শ	ণ	1.21
ৃ	ু	3.22	ণ	শ	2.17
ম	য়	1.49	ন্ত	ন্ত	11.13
য়	ম	2.11	ন্ত	ন্ত	9.14
গ্ম	গ্ম	7.49	জ্ঞ	স্ম	6.17
গ্ম	গ্ম	5.33	স্ম	জ্ঞ	8.37

Figure 2.15: Some confusing character pairs in Bangla and their percentage of misclassification

2.8 Grammar Check

Greater accuracy can be acquired if grammar of the language being scanned is taken into consideration. It helps determine whether the word is more likely to be a noun or a verb. A grammar checker is implemented either as a program or an integral part of a program and it is used to check the grammatical soundness of the written text in the context of that particular language. It is generally observed that a grammar checkers are implemented as a part of a larger program. But stand alone programs are also available.

Natural processing of a language is used for the implementation of a grammar checker [15] . Collection of idiosyncratic errors is a formation of a large part grammatical definitions. This basically depends on the language of that specific system. And also of the knowledge of that user on that particular language. The usefulness of this system can be fully realized after it is integrated to a word processing software. A grammar checker checks the syntactic soundness of a sentence or some portion of it and notifies if any corrections are required to be made. Preferably the system will also provide some linguistic explanation about what actually is wrong with that sentence or that particular portion of it. Neither excessive noise nor silence is appreciated because we wouldn't want any false alarms or an error going unnoticed. After the check for grammatical correctness is done, the accuracy of the formed sentences increases.

2.9 Pattern Matching

Checking an obtained sequence of tokens to check if the constituents of patterns are present or not is known as pattern matching [15] . Unlike pattern matching where the patterns of a letter gets checked, here the sequence of tokens gets tested and unlike pattern recognition the match generally has to be near exact. Patterns generally are stored in a sequence or in the form of a tree. Notable applications of pattern matching include finding out the position of a pattern in a given sequence of tokens. Also the system is used to output specific components of a sequence of tokens or replacing it with another token sequence. Obtaining labels or class codes of a character pattern is the final aim of this system. The main task of recognition is taking a word or character pattern and mapping them to class sets that are predefined from earlier tests and the training period.

2.9.1 Bayes Decision Theory

This refers to a decision theory that is informed beforehand by Bayesian Probability [21]. This statistical system attempts at quantifying the chances among several possible decisions. It makes use of the costs and probabilities that it has acquired during the training steps. Bayesian decision theory implies that the distribution of any kind of probabilities represents a previous distribution.

Assume that there are d feature measurements x1, . . . , xd which is extracted from the input pattern. Then the pattern is represented by a d dimensional feature vector x = [x1, . . . , xd]T . We consider that x belongs to to one of the M predefined classes 1, . . . , M. Given the a priori probabilities P(i) and class conditional probability distributions p(x—i), i = 1, . . . , M, the a posteriori probabilities are computed by the Bayes formula [15] .

2.10 Knowledge Driven Validation

A knowledge driven analyzer is used to carry out this particular validation process. For the optimization of the analyses text in a specific context a brute force searching is carried out aided by an efficient search engine. The occurrence of certain string patterns withing a larger text is found out using the string searching algorithms. This algorithms, which are also known as string matching algorithms are an important class of string algorithms [15] . Strings that are encoded can considerably effect the speed and efficiency of the algorithms. The use of a variable width encoding reduces slows the algorithms down to a great extent. It is much easier if we use a set of strings and then pass the complete set through the algorithm. The implementation of text as a set of strings makes the job easier for the search engines. This consequently affects the search percentage which results in greater percentage of accuracy as well as efficiency of the OCR engine.

2.10.1 Boyer-Moore String Searching Algorithm

The Boyer-Moore algorithm for searching strings is considered to be the the most efficient among the practical string search literature. The string that is being searched for the pattern is preprocessed by this algorithm. But it leaves out the strings that are searched in the text. As a result it produces the optimum results where the text is considerably longer than the pattern or in cases where the pattern does not change even across numerous searches. This particular algorithm uses the information that is achieved during the pre-processing steps so that it can skip portions of the text. This results in lower constant factors when compared to the other string algorithms making this a benchmark among its peers. The length of the pattern plays a great role in determining the speed of this algorithm as the speed increases

along with the length of the pattern. The checking of context of the strings that have been recognized lends certainty about the context that the engine recognized.

CHAPTER 3
METHODOLOGY

3.1 Initial scenario of our project

From the side of time, Bangla is a recent past when the research for this language has been started. In this part of the world, Natural Language engineering is a much more new story. So the advancement in computer technology took longer to have noticeable effect in this region than the western world. At first it was main concern to recognize the Bangla characters optically, the fifth-most popular language in the world. But at the time of doing that some error occurred. For this automatic error detection and correction have become a great challenge for present days. Two distinct error categories are categorized for word errors, namely, non-word errors and real word errors. Non-words mean invalid words and real word error means a valid word but not the perfect one for the sentences by making the sentence syntactically or semantically ill-formed or incorrect. Thus in both cases, the main target is to detect the word error either using suggestion by drop down or automatically replace it with an appropriate valid word. In our research field we assume that all sentences are well formed so we only consider the non-word errors. As it is already told that we faced some problem for doing OCR. One reason of that is complex character grapheme structure of Bangla script creates difficulty in error detection and correction. In bangle alphabet, there are 11 vowels and 39 constants, which known as Basic characters. The basic characters are shown in Figure 3.1.

Figure 3.1: Basic Characters of Bangla Alphabet (Vowels)

But the vowels, depending on their position in a word, take different shapes, called modifiers. Compound characters are formed by two or more basic characters.

Normally error can be detected in two ways. 1) n-gram verification and 2) dictionary look-up [12] . In the way of dictionary look-up, it was the main concern that to identify the each letter. But the main problem was to identify the compound characters in the text. Other

ক	খ	গ	ঘ	ঙ
চ	ছ	জ	ঝ	ঞ
ট	ঠ	ড	ঢ	ণ
ত	থ	দ	ধ	ন
প	ফ	ব	ভ	ম
য	র	ল	ব	শ
ষ	স	হ	ড়	ঢ়
য়	ৎ	ং	ঃ	ঁ

Figure 3.2: Basic Characters of Bangla Alphabet (Consonants)

Vowel	আ	ই	ঈ	উ	ঊ	ঋ	এ	ঐ	ও	ঔ
Modified Shape	া	ি	ী	ু	ূ	ৃ	ে	ৈ	ো	ৌ
When attached to constant ক	কা	কি	কী	কু	কূ	কৃ	কে	কৈ	কো	কৌ

Figure 3.3: Examples of mModifiers in Bangla

alphabet can be identified easily than the compound characters. In OCR system, the text is scanned first and then the alphabets are recognized through some predetermined processes. While scanning the written text, if any alphabet can not be recognized properly then the alphabet is replaced by a group of letters (cluster).

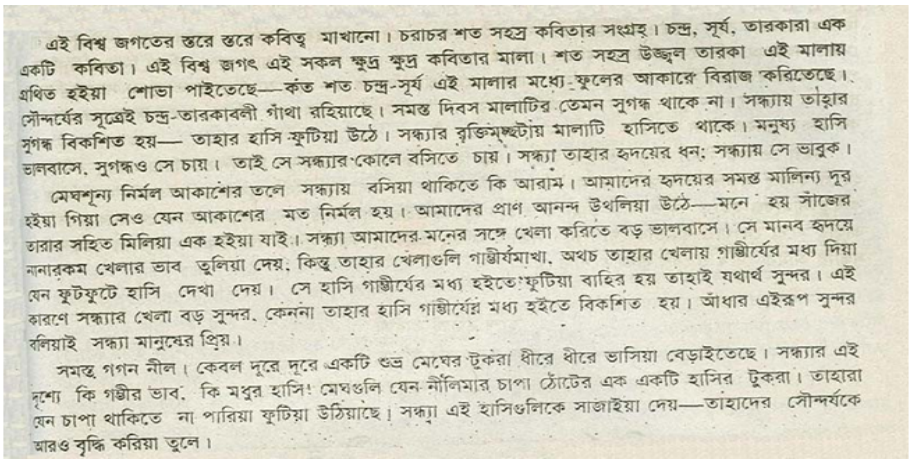
This type of bangla text image is converted into a grayscale format first then it is converted into some connected components.

3.2 Lexicon Formation

Thus this type of problem occurred, some new methods or processes were needed to solve this problem. For this, as an initiative step, the words were searched from a text file or word dictionary according to our given letter. The word is searched according to the given letters

Constant characters		Compound characters
ক + র	=	ক্র
ক + ষ	=	ক্ষ
গ + ধ	=	গ্ধ
চ + চ	=	চ্চ
শ + চ	=	শ্চ
ক + ষ + ণ	=	ক্ষ্ণ
ন + ত + ব	=	ন্ত্ব
ন + ত + র	=	ন্ত্র

Figure 3.4: Examples of Compound Characters in Bangla



এই বিশ্ব জগতের স্তরে স্তরে কবিত্ব মাখানো। চরাচর শত সহস্র কবিতার সংগ্রহ। চন্দ্র, সূর্য, তারকারা এক একটি কবিতা। এই বিশ্ব জগৎ এই সকল ক্ষুদ্র ক্ষুদ্র কবিতার মালা। শত সহস্র উজ্জ্বল তারকা এই মালায় গ্রথিত হইয়া শোভা পাইতেছে—কত শত চন্দ্র-সূর্য এই মালার মধ্যে ফুলের আকারে বিরাজ করিতেছে। সৌন্দর্যের সূত্রেই চন্দ্র-তারকাবলী গাঁথা রহিয়াছে। সমস্ত দিবস মালাটির তেমন সুগন্ধ থাকে না। সন্ধ্যায় তাহার সুগন্ধ বিকশিত হয়— তাহার হাসি ফুটিয়া উঠে। সন্ধ্যার রক্তিমচ্ছটায় মালাটি হাসিতে থাকে। মনুষ্য হাসি ভালবাসে, সুগন্ধও সে চায়। তাই সে সন্ধ্যার কোলে বসিতে চায়। সন্ধ্যা তাহার হৃদয়ের ধন; সন্ধ্যায় সে ভাবুক।

মেঘশূন্য নির্মল আকাশের তলে সন্ধ্যায় বসিয়া থাকিতে কি আরাম। আমাদের হৃদয়ের সমস্ত মালিন্য দূর হইয়া গিয়া সেও যেন আকাশের মত নির্মল হয়। আমাদের প্রাণ আনন্দ উথলিয়া উঠে—মনে হয় সাঁজের তারার সহিত মিলিয়া এক হইয়া যাই। সন্ধ্যা আমাদের মনের সঙ্গে খেলা করিতে বড় ভালবাসে। সে মানব হৃদয়ে নানারকম খেলার ভাব তুলিয়া দেয়; কিন্তু তাহার খেলাগুলি গাম্ভীর্যমাখা, অথচ তাহার খেলায় গাম্ভীর্যের মধ্য দিয়া যেন ফুটফুটে হাসি দেখা দেয়। সে হাসি গাম্ভীর্যের মধ্য হইতে ফুটিয়া বাহির হয় তাহাই যথার্থ সুন্দর। এই কারণে সন্ধ্যার খেলা বড় সুন্দর, কেননা তাহার হাসি গাম্ভীর্যের মধ্য হইতে বিকশিত হয়। আঁধার এইরূপ সুন্দর বলিয়াই সন্ধ্যা মানুষের প্রিয়।

সমস্ত গগন নীল। কেবল দূরে দূরে একটি শুভ্র মেঘের টুকরা ধীরে ধীরে ভাসিয়া বেড়াইতেছে। সন্ধ্যার এই দৃশ্যে কি গম্ভীর ভাব, কি মধুর হাসি! মেঘগুলি যেন নীলিমার চাপা ঠোঁটের এক একটি হাসির টুকরা। তাহারা যেন চাপা থাকিতে না পারিয়া ফুটিয়া উঠিয়াছে। সন্ধ্যা এই হাসিগুলিকে সাজাইয়া দেয়—তাহাদের সৌন্দর্যকে আরও বৃদ্ধি করিয়া তুলে।

Figure 3.5: An Example of Digital Image of Bangla Text

of the words. There can be some letter missing in the middle, front or back position in the word. But the word is searched by matching the string from the given word file. From the drop down option the required word can be found out.

For example, if we want to search any word by letter and then we type the letter then we will find out the expected word from the drop down option if this word is stored in the text file. Figure 3.7 is the example of the this.

3.3 Training The System

For predicting words first we had to separate all the words from the text. For doing this separation perfectly, we show a new word with frequency 1. If we get any word which we have got earlier then we will just increase the frequency of the word, no new word will be shown for that.

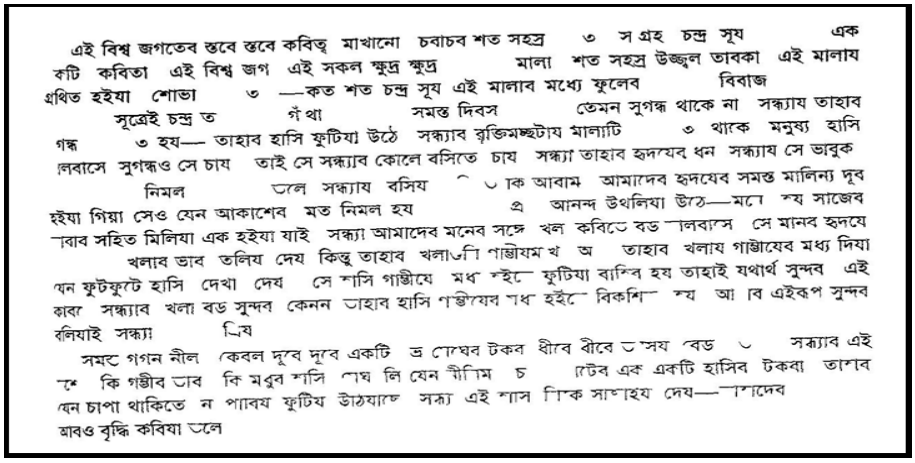


Figure 3.6: Selected Connected Components

Search by	Drop down options
ম	মা, মাতৃভূমি, মাতৃভাষা
মা_ভ	মাতৃভূমি, মাতৃভাষা

Figure 3.7: Example of Search Results

3.4 Formation of Root Word Lexicon

For predicting the unknown or error word, a new method is applied to our project. Besides the separated current word we also store the previous and post word of the current word with their frequencies. It helps to find out that which word has more possibility to be with the current word. Actually the main concept is to find out the correct word instead of the error word with the help of the post and previous word of the current word. Following is the details process or concept of the whole process.

While find out the previous and post word we use a list named Lexicon. The lexicon is constructed with a list of previous word, a list of post word, an integer for recording the frequency of the current word and a string named Current where the current word is stored. In Figure 3.9 the structure of the lexicon is shown.

Here we first find out the current word whether it is listed previously or not. If not then it will be listed and its frequency will be also set. Then the previous word of that word will be listed in previous word list and the post word will be in post word list with their frequency of that current word. And if the current word is stored previously then no new word is stored in lexicon. In that case, there a search takes place whether the previous/post word is stored earlier for that current word or not. If they not then they are stored in previous/post list otherwise the frequency of that corresponding previous or post words frequency is increased.

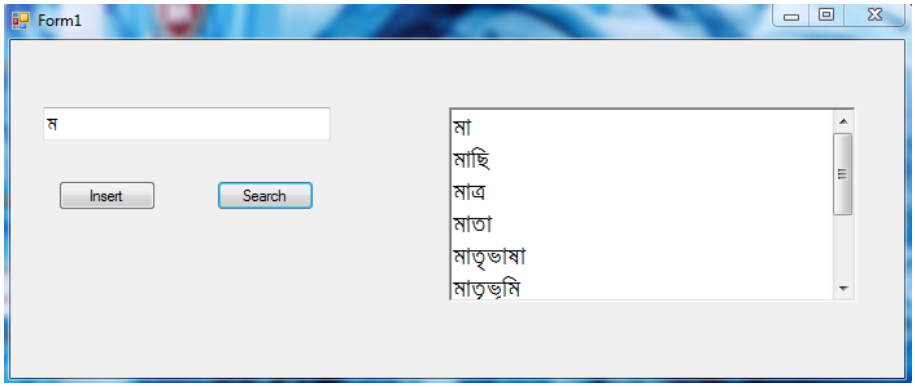


Figure 3.8: First Step of Our Project

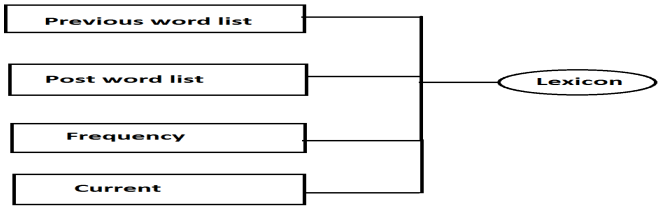


Figure 3.9: Structure of Lexicon

3.5 Formation of Previous/Post Word

The structure of previous and post word list is quite similar to the lexicon. In previous post word list, the components are the string of the word and the frequency of the word. In this case, similar to lexicon, if we get a new previous or post word for the current word then we insert a new word. Otherwise the frequency of the word is just increased. In Figure 3.10 the structure is shown.

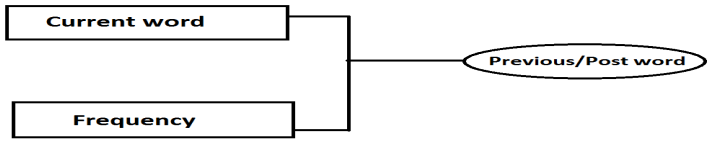


Figure 3.10: Structure of Previous/Post Word

3.6 Text File Implementation of Lexicon

By doing so a text file can be divided into current word and its previous and post word list and their frequencies. In Figure 3.11, a text is written in bangle Unicode and in Figure 3.12 this text is separated into current words, previous words and post words.

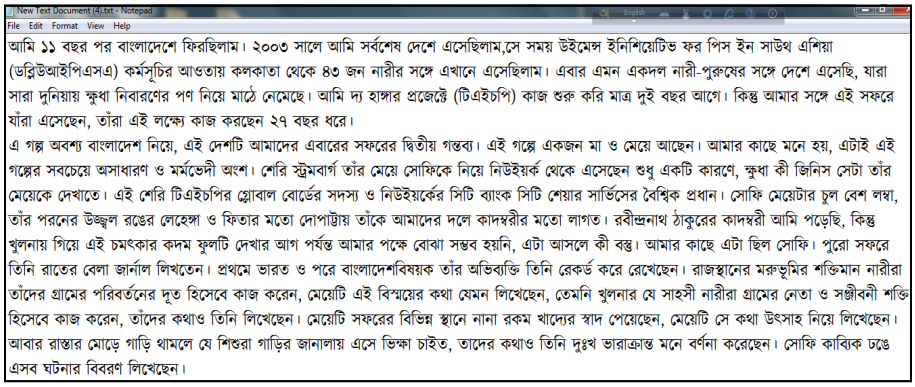


Figure 3.11: Text File for Processing

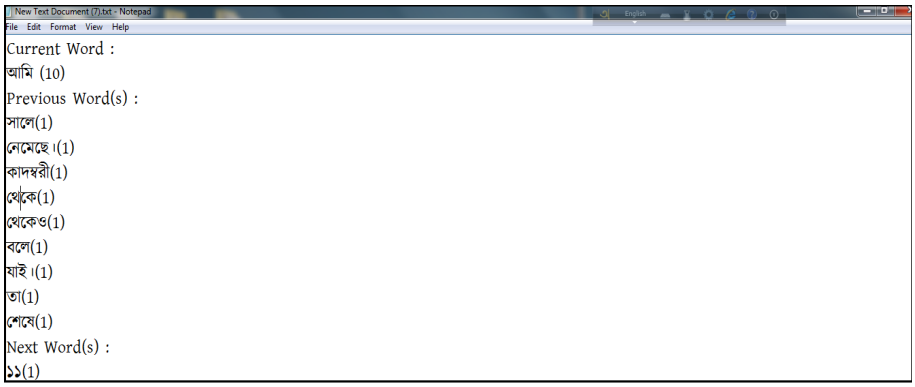


Figure 3.12: Processed Text File

3.7 Database Implementation of Lexicon

An important aspect in the usability of a Lexicon is its (technical) appearance. Many small, experimental systems have been built on stand-alone personal computers, using tools that were quite appropriate for the scientific project targets but not to support practical applications in a production environment. The Lexicon design as described in this paper is aimed at large-scale collaborative production, providing multi-user access and high performance on an appropriate platform. It can be integrated in external applications when used as a network server, and just as with traditional database systems, will be shielded from casual or non-technical / non-linguist users with appropriate front ends [22] . It can be applicable for a specific type of file that means a fixed context type database or text file as Bangla language is a vast area for predicting misspelled words.

Bangla lexicon development can be described as the continuous interaction of three layers of functions. They are depicted below:

The Model View Controller (MVC) architecture is main structure in lexicon framework development. The MVC has three parts: the model, the view, the controller. These three can be considered as the processing, the output, the input. Input–>Processing–>Output

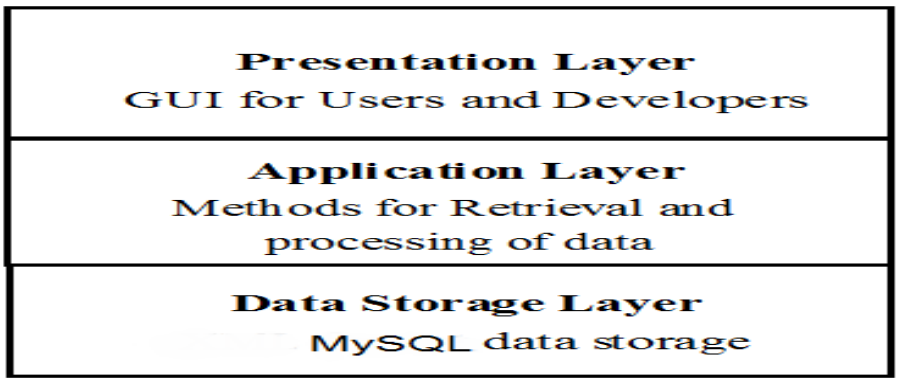


Figure 3.13: Three Layer Function

Controller –> Model –> View

The user input, the modeling of the external world, and the visual feedback to the user are separated and processed by model. The controller interprets mouse and keyboard inputs from the user and maps these user actions into commands that are sent to the model. The model manages one or more data elements, responds to queries about its state, and responds to instructions to change state [22] .

The model in the lexicon development project consists of the MySQL containing lexicon data and the c# which will be used to process user input and output formatting. The lexicon development interface will have a web based view and a standalone view. Both the view will be used for entering data into the lexicon. The web based interface can be used by all people, so the data that is entered through the web interface will not be directly entered into the final lexicon file. Instead they will be stored into a primary MySQL for further validation for correctness and redundancy by specialists. After correction the relevant data would be stored into the final MySQL lexicon file and hence the lexicon will be updated. The intermediate web technology will be controlling the interaction of the model and view components. The standalone interface is a form based interface that allows a user to enter necessary data directly into a MySQL file. In our database there are three tables namely word, prevword, postword. Word has the structure of the lexicon which has been shown before in the structure of lexicon and prev/post words (Figure 3.9, Figure 3.10). In Figure 3.14 the table of our database is shown:

3.8 Detection of Faulty Word

In presentation layer we put the input constructed with faulty word. Then at first the faulty word is marked among it. Then we search and find out the previous and post word of the faulty word. According to the frequency and comparability the most used and possible is find out. If there is a tie or complexity to find out the most used word then we discuss about

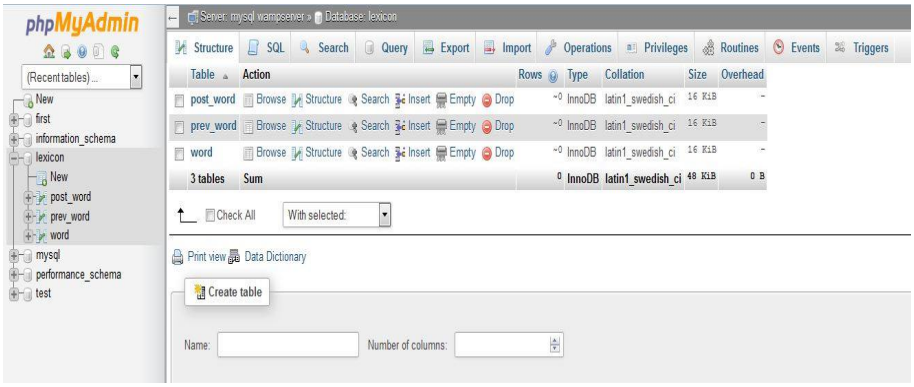


Figure 3.14: Database of Our Project

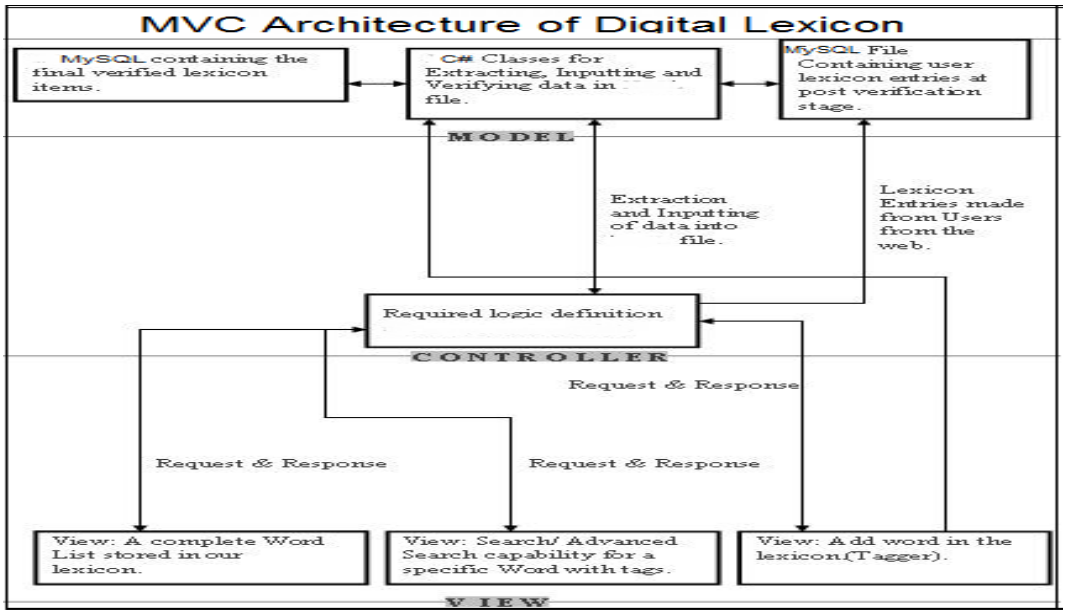


Figure 3.15: MVC Architecture of Digital Lexicon

the previous to previous word of the faulty word and same as post words. In Figure 3.16 a text file is shown with faulty words and in Figure 3.17 the result of finding faulty words in out project is shown.

In this case the faulty letter of the misspelled word was indicated by *. But if misspelled word is not marked then we will identify the misspelled word by searching each word of document in our database. We will match the letter of the words of the document with the words of the database. Then according to the matching level of two strings we gave every word a confidence percentage. If any words confidence level crosses the threshold value (assumed) then it will be considered as faulty or misspelled word.

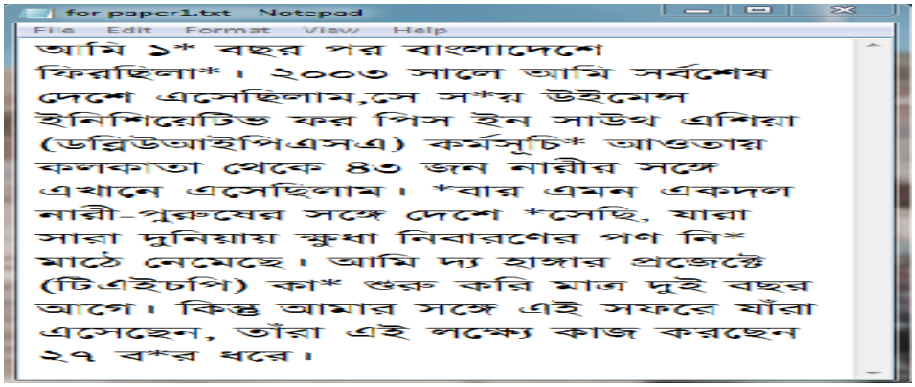


Figure 3.16: A Text File with Faulty Words

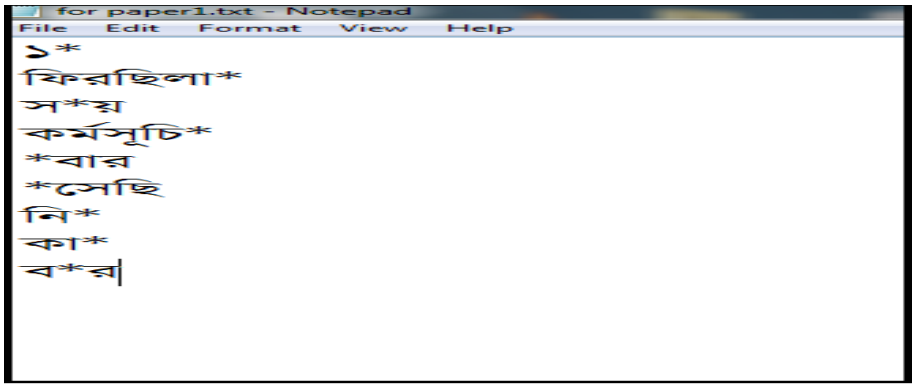


Figure 3.17: Results of Finding Faulty Words from The Text

3.9 Correction of Faulty Word

After finding the faulty words, based on the previous and post words the correct word van be found out. If there arises more complexity to find out this just based on the immediate previous and post words then the more backward process is required. After applying this process we can find out the faulty words thus reach nearly to our goal.

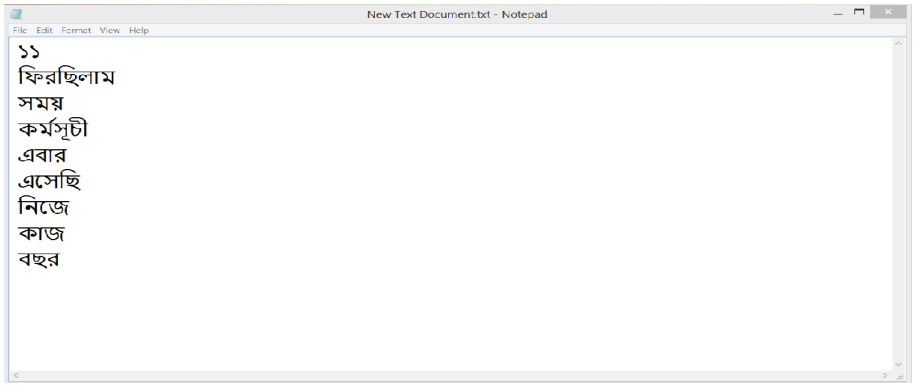


Figure 3.18: Corrected Words List

The whole procedure can be described through a framework [23] . In Figure 3.19 the total

framework is shown-

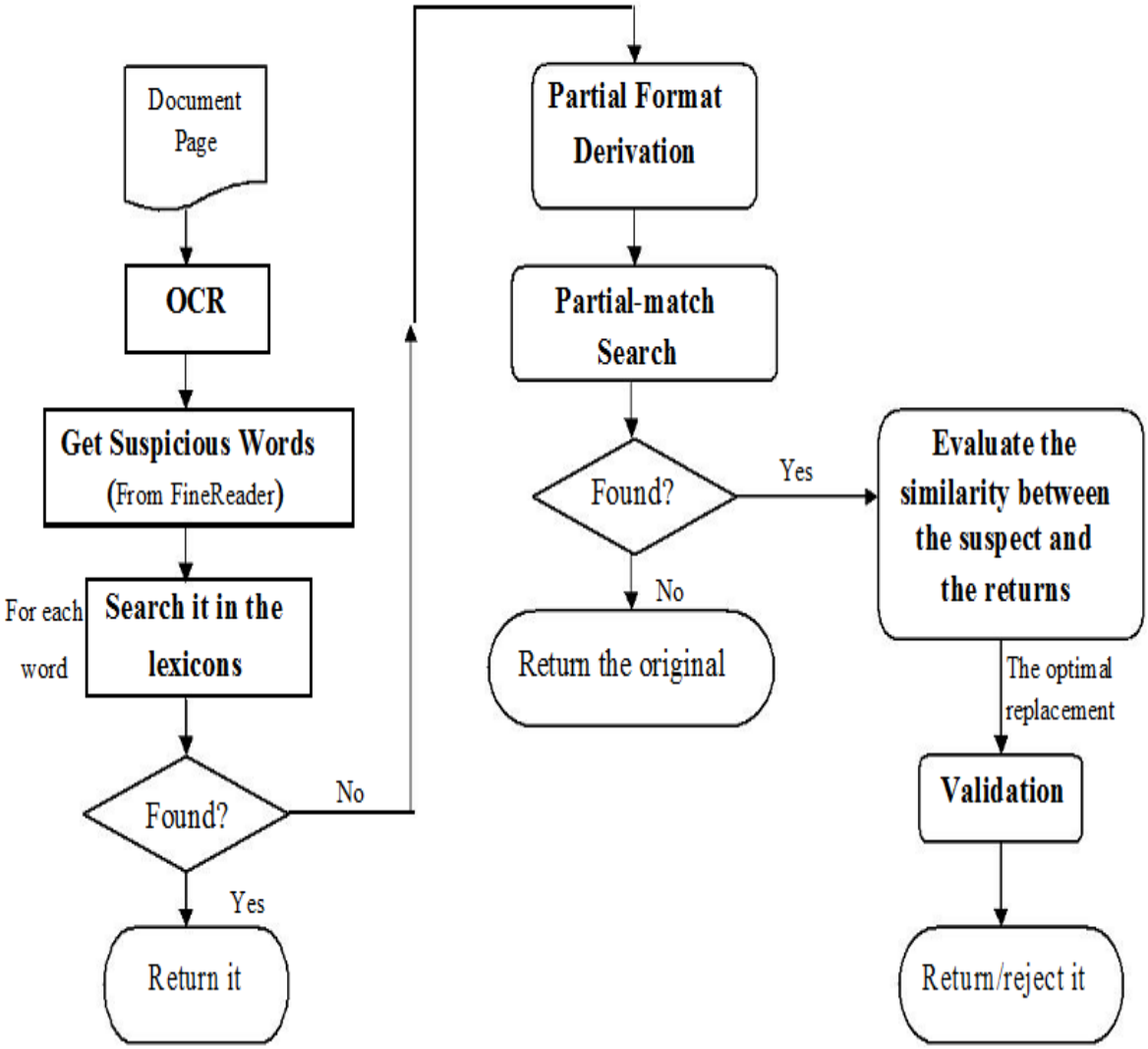


Figure 3.19: Framework of The Process [23]

CHAPTER 4
RESULTS AND ANALYSIS

4.1 Experiment the results

There are many spell checker is available for checking the spelling of the misspelled words of different languages. For bangla, bangle word checker is designed such a way to get the higher efficiency. So for testing the ability of correcting the word errors, we conducted several experiments. As it has been mentioned before that our system always work for a fixed type of documents at a time, based on the lexicon word type as Bangla is a vast area for predicting words or correcting misspelled words depending on the prefix and suffix of the word. Following are some cases showing the screen shot and result tables for that several cases.

Case-1: Documents related to Sports (Cricket)

Here in Figure 4.1 a document is shown which is related to International Cricket.

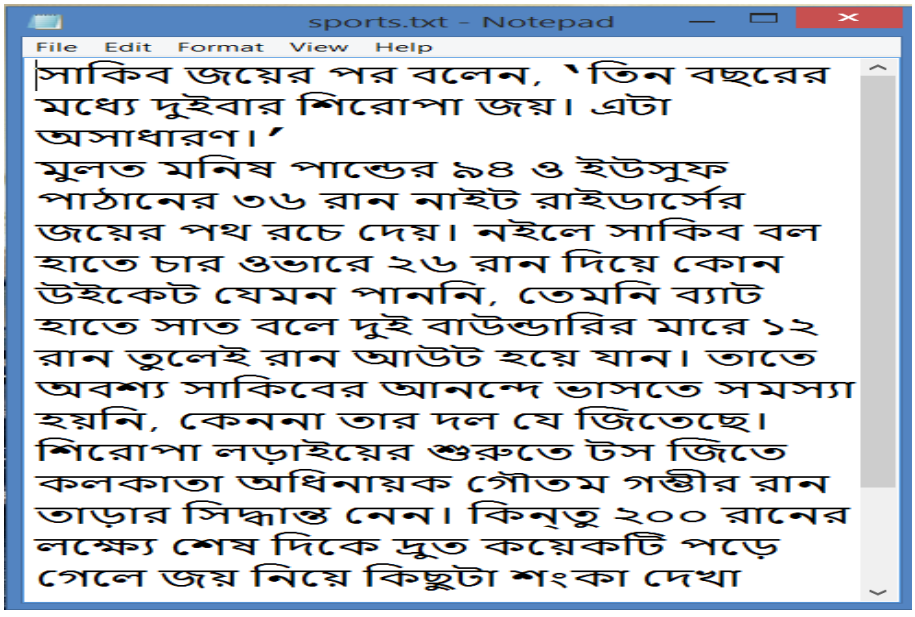


Figure 4.1: Text Document about International Cricket

Then in Figure 4.2 some error or misspelled word is given for testing the system. As our concern was international cricket so we stored words related this to our database. There were 9 misspelled words. In Figure 4.3 we can see that our system corrected 8 words among

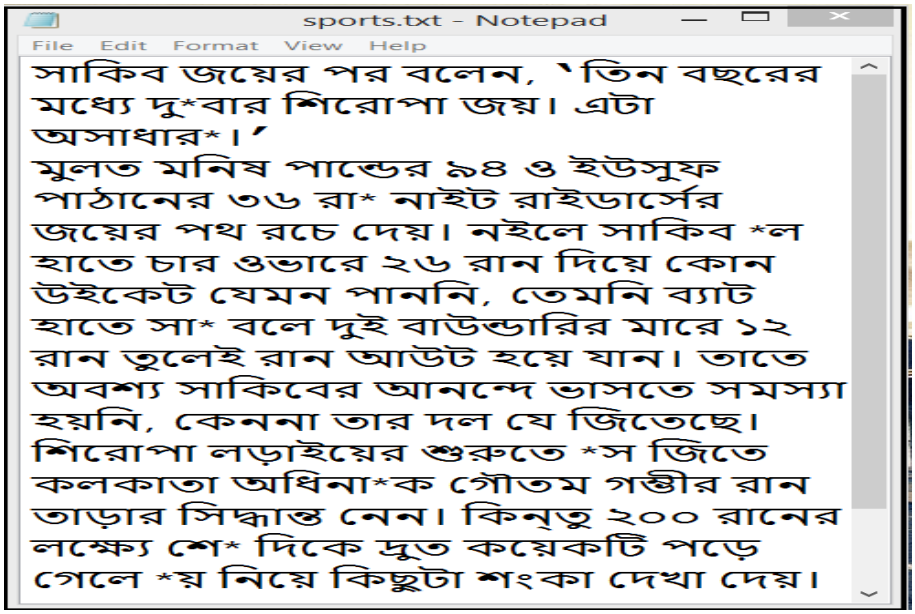


Figure 4.2: Text Document with Misspelled Words of Fiugre 4.1

them and 1 word remained misspelled.

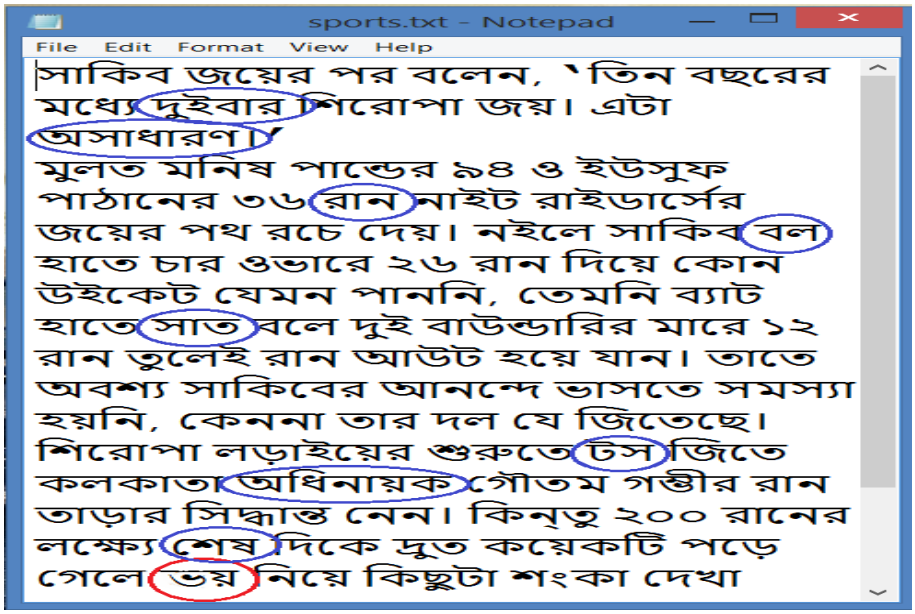


Figure 4.3: Corrected Text File by the System of Fiugre 4.2

In Figure 4.3 the blue marked words are correctly predicted by the software while the red signed word was not predicted correctly. Later a Table 4.1 is given showing the accuracy percentage of our system.

Case-2: Documents related to Politics Here in Figure 4.4 a document is shown which is related to Politics.

Then in Figure 4.5 some error or misspelled word is given for testing the system. As our

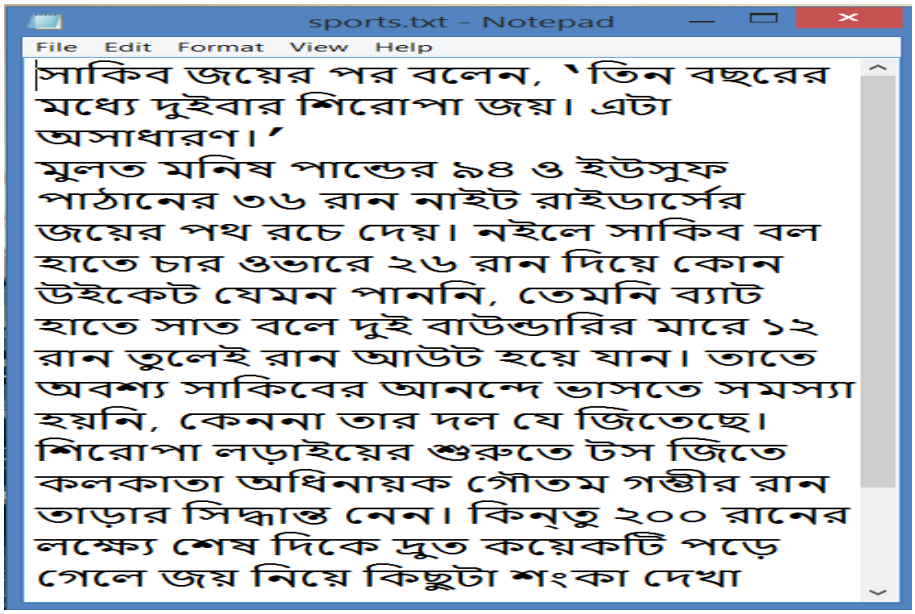


Figure 4.4: Text Document about Politics

concern was politics so we stored words related this to our database. There were 11 misspelled words. In Figure 4.6 we can see that our system corrected 8 words among them and

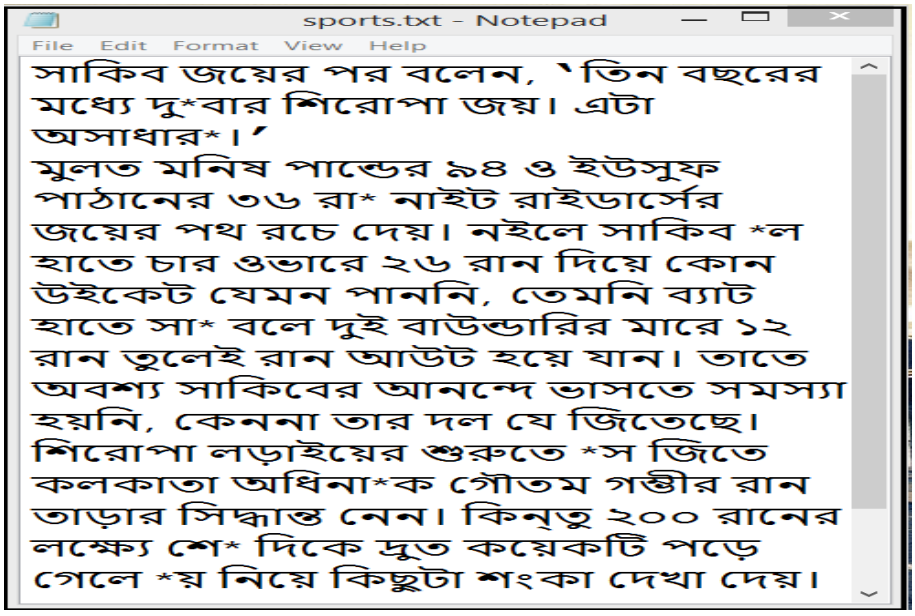


Figure 4.5: Text Document with Misspelled Words of Fiugre 4.4

1 word remained misspelled.

In Figure 4.6 the blue marked words are correctly predicted by the software while the red signed word was not predicted correctly. Later a table 4.1 is given showing the accuracy percentage of our system.

Case-3: Documents related to International Affairs Here in Figure 4.7 a document is

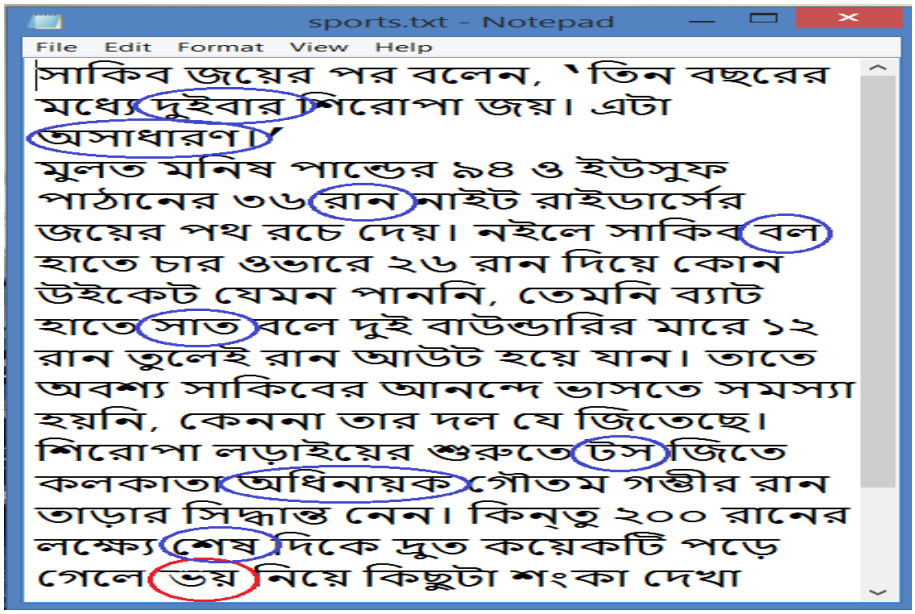


Figure 4.6: Corrected Text File by The System of Fiugre 4.5

shown which is related to International Affairs.

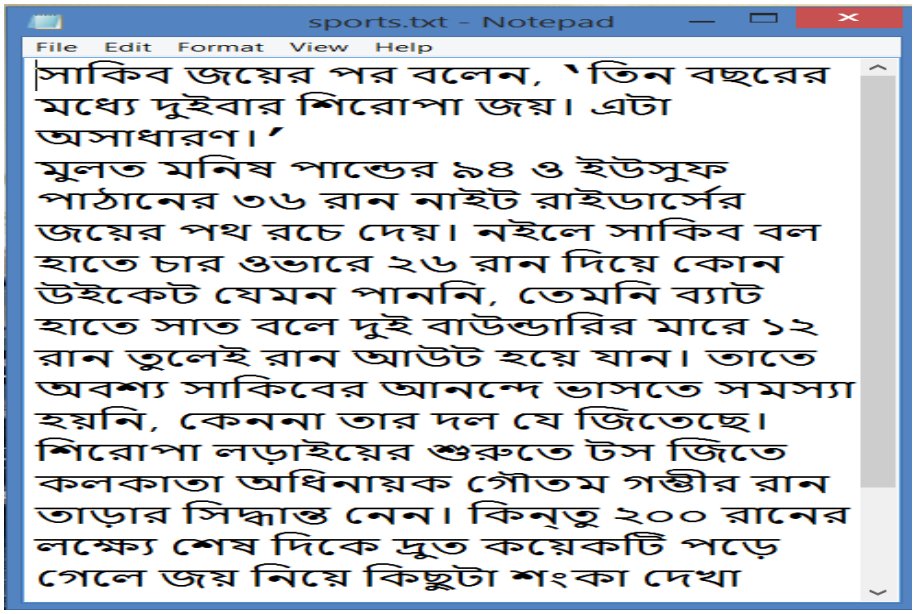


Figure 4.7: Text Document about International Affairs

Then in Figure 4.8 some error or misspelled word is given for testing the system. As our concern was politics so we stored words related this to our database. There were 9 misspelled words.

In Figure 4.9 we can see that our system corrected 8 words among them and 2 word remained misspelled.

In Figure 4.9 the blue marked words are correctly predicted by the software while the red

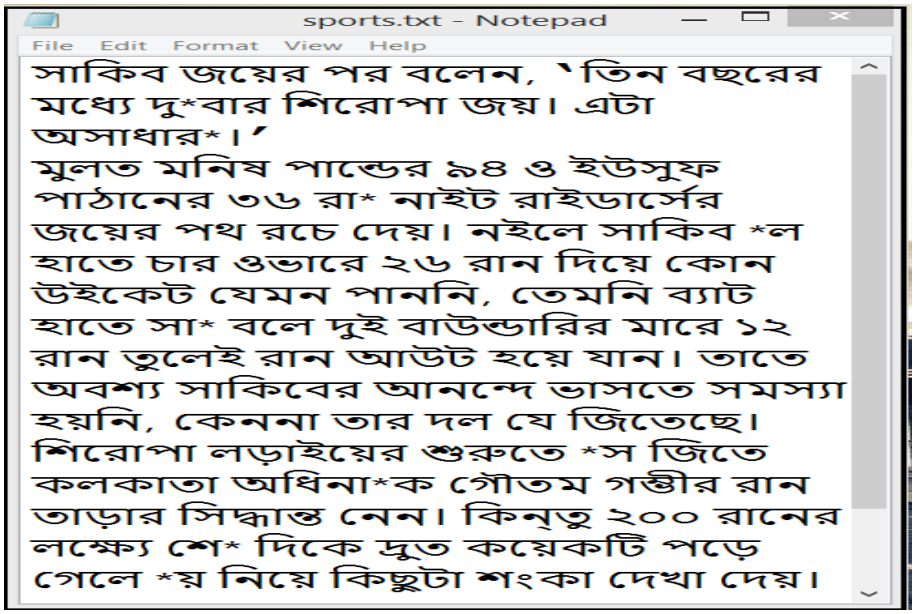


Figure 4.8: Text Document with Misspelled Words of Fiugre 4.7

Table 4.1: Accuracy percentage of our system based on the cases

Documents on	Number of Misspelled Word	Number of Corrected Words	Accuracy %
Sports (Cricket)	9	8	88.89%
Sports (Football)	9	9	100%
Politics	11	10	90.09%
International Affairs	7	5	71.43%

signed word was not predicted correctly. Later a Table 4.1 is given showing the accuracy percentage of our system.

Case-4: Documents related to International Sports (Football) Here in Figure 4.10 a document is shown which is related to International Sports (Football).

Then in Figure 4.11 some error or misspelled word is given for testing the system. As our concern was politics so we stored words related this to our database. There were 7 misspelled words. In Figure 4.12 we can see that our system corrected all the 7 words.

In Figure 4.12 the blue marked words are correctly predicted by the software while the red signed word was not predicted correctly. Later a table 4.1 is given showing the accuracy percentage of our system. Following is the table calculating the accuracy percentage of the system:

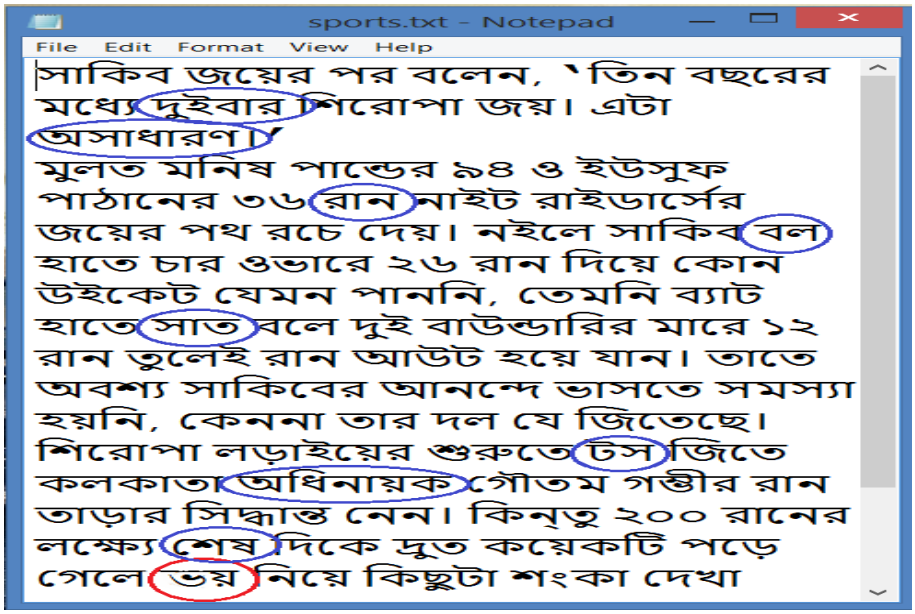


Figure 4.9: Corrected Text File by The System of Fiugre 4.8

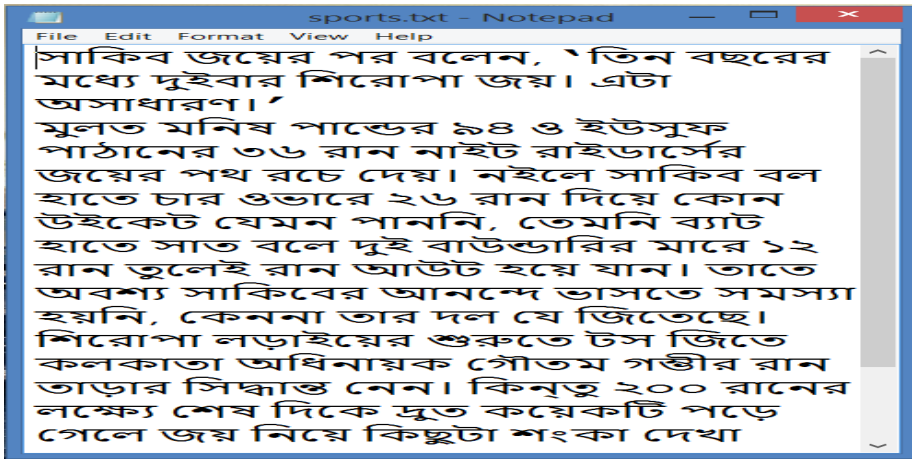


Figure 4.10: Text Document about International Sports (Football)

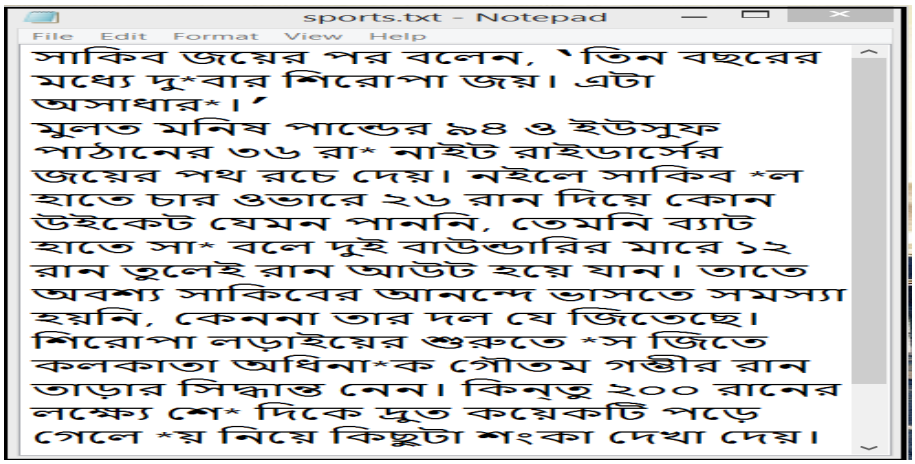


Figure 4.11: Text Document with Misspelled Words of Fiugre 4.10

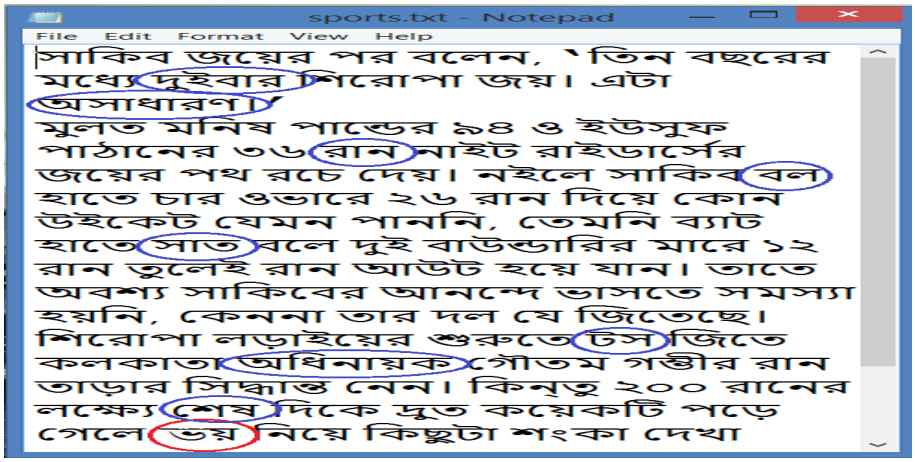


Figure 4.12: Corrected Text File by The System of Fiugre 4.11

CHAPTER 5
CONCLUSION

Although Bangla is one of the most popular languages in the world, research regarding the processing of this language is still largely unsatisfactory. To keep pace with the ever advancing technology, digitization of Bangla language is of utmost importance. If we fail in this task, it will be very difficult to maintain the relevance & popularity of the language Bangla in the future. Its a matter of great hope and inspiration that this particular field meaning Bangla language development is now a priority project of our government also. Government has taken initiative by incorporating properly funded projects in this regard. Development of an accurate and efficient spell checker will go a long way in ensuring proper, efficient and accurate digitization of Bangla language. But for issues discussed in the above sections, it is very complicated to develop a Bangla spell checker if the correction method is based on the errors committed on the character level. Thus, it is our objective to develop a Bangla spell checker whose correction methodology is based on words, with the aid of a unique kind of lexicon. The success of our endeavor will contribute greatly in correcting the misspelled words in text files which in turn will be a great boon for the digitization systems being developed for the Bangla language.

5.1 Limitation of the system

As every system has lackings, our system is not exceptional of that. Bangla is a vast language so many of the times the consequences of the words depend on the sepecific type of the document. For this, at a time a specific type of documents can be processed by our system. Another limitation of our system is that it can not predict two successive misspelled words. We are predicting words in our system with the help of the previous and post words of the misspelled words. Thus in our system it is impossible to correct any two successive faulty words. In our system we have to conduct with a relatively large lexicon. A more shopisticated algorithm can be used for this system.

5.2 Future expansion of the system

The proposed system here is just concern about the spelling of the word comparing with just the previous and post words of the word. In future it can be expanded as to predict the words with the help of the meaning of the sentences. Besides more extensive evaluations will provide the statistical information needed to manage the suffix list, which in turn will determine the trade-off between under-stemming and over-stemming. And now we are working just with bengla language. Hope near future this can help to predict the Bengalis sister languages such as Assamese and Oriya.

REFERENCES

[1] B. S. M. H. R. Nur Hossain Khan, Gonesh Chandra Saha, "Checking the correctness of bangla words using n-gram(975-8887)," *Computer Applications*, vol. 89, no. 11, 2014.

[2] M. K. Naushad UzZaman, "A comprehensive bangla spelling checker,"

[3] M. K. Md. Zahurul, Md. Nizam Uddin, "A light weight stemmer for bengali and its use in spelling checker,"

[4] W. H. W. Kyongho Min, "Syntactic recovery and spelling correction of iii-formed sentences,"

[5] "Natural language processing." Last accessed on December 17, 2014, at 02:08:00PM. [Online]. Available: http://en.wikipedia.org/wiki/Natural_language_processing.

[6] "Information retrieval." Last accessed on December 16, 2014, at 02:08:00PM. [Online]. Available: http://en.wikipedia.org/wiki/Information_retrieval.

[7] "Information extraction." Last accessed on December 16, 2014, at 02:08:00PM. [Online]. Available: http://en.wikipedia.org/wiki/Information_extraction.

[8] "Natural language generation." Last accessed on December 17, 2014, at 02:08:00PM. [Online]. Available: http://en.wikipedia.org/wiki/Natural_language_generation.

[9] "Speech recognition." Last accessed on December 17, 2014, at 02:08:00PM. [Online]. Available: http://en.wikipedia.org/wiki/Information_extraction.

[10] "Optical character recognition." Last accessed on December 16, 2014, at 02:08:00PM. [Online]. Available: http://en.wikipedia.org/wiki/Information_extraction.

[11] K. F. John Evershed, "Correcting noisy ocr: Context beats confusion."

[12] Y.-H. Tseng, "An approach to retrieval of ocr degraded text," *Library Science*, no. 13, pp. 1018–3817, 1998.

[13] P. M. Neha Gupta, "Spell checking techniques in nlp," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 12, 2012.

[14] M. K. Md. Tamjidul Hoque, "Coding system for bangla spell checker," *A Survey*, no. 12, 2013.

[15] D. S. P. Y. Piyush Sudip Patel, "Post-processing on optical character recognition," *IJAEA*, vol. 2, no. 6, pp. 71–76, 2009.

[16] "Digital search tries." Last accessed on December 14, 2014, at 02:08:00PM. [Online]. Available: www.lsi.upc.edu/.

[17] B. C. U. Pal, P.K. Kundu, "Ocr error correction of an inflectional indian language using morphological parsing," *Information Science And Engineering*, no. 16, pp. 903–922, 2000.

[18] S. E. H. Thomas A. Laskoa, "Approximate string matching algorithms for limited-vocabulary ocr output correction," no. 16, 2000.

[19] N. K. Ritika Mishra, "A survey of spelling error detection snd correction techniques," *Computer Trends and Technology*, vol. 4, no. 3, 2013.

[20] L. E. N. S. A.C. Jobbins, G. Raza, "Post processing for ocr: Correcting errors using semantic relations,"

[21] "Bayesian decision theory." Last accessed on December 13, 2014, at 12:08:00AM. [Online]. Available: www.wikipedia.org/.

[22] F. M. S. D. M. K. Dewan Shahriar Hossain Pavel, Asif Iqbal Sarkar, "Collaborative lexicon development for bangla," no. 12, pp. 1–7, 2014.

[23] D. M. Siyuan Chen and G. R. Thoma, "Efficient automatic ocr word validation using word partial format derivation and language model," *Soft Computing*, vol. 14, no. 12, pp. 1329–1337, 2010.