TABLE OF CONTENTS

EDITORIAL ADVISORY BOARD

DISCLAIMER

All the contributions are published in good faith and intentions to promote and encourage research activities around the globe. The contributions are property of their respective authors/owners and the journal is not responsible for any content that hurts someone's views or feelings etc.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
VOLUME 5, NO. 1
ISSN 2054-7390

Construction of Word Dictionary for Bangla Vowel Ended Roots and Its Verbal Inflexions in UNL Based Machine Translation Scheme

Md. Nawab Yousuf Ali[1], Md. Shamsujjoha[2], Golam Sorwar[3] and Shamim H Ripon[4]
[1,2,4]*Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh*
[3]*School of Business and Tourism, Southern Cross University, Gold Coast, Australia*
nawab@ewubd.edu, dishacse@yahoo.com, Golam.Sorwar@scu.edu.au, dshr@ewubd.edu

ABSTRACT

This paper focuses on the development of word dictionary of Bangla vowel ended roots and their verbal inflexions for an interlingua representation called Universal Networking Language (UNL) processors. A considerable amount of work has been done on the development of Bangla morphological analysis on verbs, nouns, prefixes and suffixes for machine translation. As far as the researchers are aware, no attempts, however, have been made to integrate the previous developments on Bangla vowel ended roots and their inflexions to a concrete computational output. This paper attempts to bridge the gap on Bangla vowel ended roots and inflexions in the framework of UNL system aiming to produce a Bangla word dictionary for UNL. The paper analyzes the Bangla vowel ended roots and verbal inflexions and develops their formats in the UNL structure. Dictionary entries of all vowel ended roots and their inflexions are developed in order to generate associated verbs for sentences. Following semantic rules these verbs can be used to convert Bangla native language sentences into UNL expressions, which are then converted into required native languages using the language specific generation rules. Conversion of a Bangla language sentence into UNL expression has also been shown in this paper.

Keywords: Verb Roots, Vowel Ended Roots, Verbal Inflexion, Bangla Word Dictionary, Universal Networking Language, and Universal Words.

1 Introduction

The Universal Networking Language (UNL) [1] is an artificial language, in the form of semantic network for computers to express and exchange various information across languages. The mission of the UNL project is to allow people to access information on the Internet in their own languages [2]. Hundreds of millions of people throughout the world, with various demographic backgrounds use Internet for information communication and sharing [3]. English is arguably, though, considered as a primary vehicle for the Internet based information, presentation and delivery, understandably not all Internet users are expected to have the necessary level of English language proficiency. Knowledge and information in different languages are scattered all over the world and remain inaccessible to mostly due to non-machine representation and language barrier [4]. Translation is the means of disseminating information; however, it demands extensive effort and cost directly and/or indirectly. Though nations are becoming more

DOI: 10.14738/tmlai.51.2479
Publication Date: 15th January, 2017
URL: http://dx.doi.org/10.14738/tmlai.51.2479

interdependent and need to exchange information, language barrier hinders these progresses at individual, institutional and national levels. Knowledge sources are to be shared globally as much as possible to advance civilization [5]. To deal with the language barrier, United Nations University/Institute of Advanced Studies (UNU/IAS) conducted a review of all internationally available machine translation programs and started to devising an efficient and workable technique to develop a human language neutral meta-language for the Internet. The result of the project is Universal Networking Language (UNL) [1]. The aim of this internationally cooperative initiative is to eliminate the massive requirement of translation among languages and reduce language to language translation to one time conversion to UNL. Once information written in one language is converted into UNL, they can be shared by anyone with their own native languages [4]. In UNL framework, each native language sentence is converted into a UNL hypergraph by a tool called "Enconverter" [6] following analysis rules defined in [7]. These hypergraphs are then translated into any native language, using generation rules defined in [7], by another tool called "Deconverter" [8]. The development of language specific components, such as dictionary, analysis rules and generation rules used by Enconverter and Deconverter, are the research focus across the world.

The people in Bangladesh and three states (West Bengal, Tripura and Aaam) in India, which is about one sixth population of the world use Bangla as their first language. About one sixth population of the world is speaking in Bangla. Exchanging information and sharing knowledge globally, it is critically important to devise conversion technique(s) for Bangla language texts into UNL and vice versa. Machine translation (MT) is an approach to translating texts from one natural language to another automatically. Ali and Ali (2002) attempted to develop MT Bangla dictionaries that address the organization, contents and details of the information [9]. Saha (2005) developed low cost English to Bangla (E2B)-ANUBAD translating English text into Bangla text using both rule-based and transformation-based MT schemes along with three-level of parsing [10]. Another attempt by Uddin et. al. (2004) was to develop a statistical Bangla to English translation engine using only simple Bangla sentences that contain a subject, an object and a verb [11].

As a consequence, the development of these aspects is the major focus of this research. A rigorous study on Bangla language grammar [9-11, 13-15], verb and roots (vowel ended and consonant ended) [9-11] and morphological analysis [3, 16-20], based on their semantic structures, has also been conducted due to the relevancy with the study.

The paper extends the work on Bangla Vowel Ended Roots (VERs) for representing them into a computational approach. To prepare word dictionary of Bangla VERs and verbal inflexions (VIs), this study has conducted an in-depth analysis of various aspects, including UNL expression, UNL Attributes, Universal Words, UNL systems and specifications of EnConverter [1-8] of UNL. Among those, Universal Words and Attributes play an important role in the development of dictionary entries for any native language word. Alike any other languages, they are equally important for the development of Bangla word dictionary, enconversion and deconversion rules required for a conversion of a natural language sentence (here Bangla sentence) into a UNL expression.

The major components of this research touch upon: 1) analysis of Bangla vowel ended roots (VERs) and their verbal inflexions (VIs), 2) categorization of VERs considering the ways verbal inflexions are added with them to form verbs, 3) identification of alternative roots for them 4) outlining the formats of VERs, 5) dictionary entries of VERs, 6) outlining the formats of verbal inflexions, 7) Dictionary entries of verbal

inflexions and 8) Conversion of a Bangla text into UNL expressions. A preliminary version of the work has been published in [20].

The rest of the paper is organized as follows. Section 2 describes the structure of UNL and EnConverter. Format of UNL-Based Bangla word dictionary is presented in Section 3. Analysis of Bangla VERs and their Vis is elaborated in Section 4. This section also presents categorizations of VERs, their alternative roots and Vis and their alternative VERs. Section 5 outlines the format of word dictionary for Bangla VERs and their lexicons. Dictionary format of Vis and their lexicons are presented in Section 6. Conversion procedures of a Bangla sentence into UNL expression is shown in Section 7, while some concluding remarks and future directions are presented in Section 8.

2 Universal Networking Language (UNL)

The UNL has been defined as a digital meta-language for describing, summarizing, refining, storing and disseminating information in a machine independent and human language neutral form [1]. It represents information, i.e. meaning, sentence by sentence. Each sentence is represented as a hypergraph, where nodes and arcs represent concepts and their relations respectively. This hypergraph is also represented as a set of directed binary relations between a pair of concepts present in a sentence. Concepts are represented as character-strings called Universal Words (UWs). Knowledge in UNL document is expressed in the following three dimensions [8]:

2.1 Universal Words (UWs)

UWs, which are language independent, are used to express word knowledge. UWs constitute the UNL vocabulary and the syntactic and semantic units, which are combined according to the UNL laws to form UNL expressions. They are tagged using restrictions describing the sense of a word in a current context. For example, drink(icl>liquor) denotes a sense of drink, as a noun- restricting the sense to a type of liquor. Here, icl stands for inclusion forming an is-a relation as in semantic nets

2.2 Relation Labels (RL)

Conceptual knowledge is captured by the relationship between UWs through a set of UNL relations. For example, Human affects the environment is described in UNL expression as:

agt (affect(icl>do).@present.@entry:01,human(icl>animal).@pl)

obj(affect(icl>do).@present.@entry:01,environment (icl>abstract

thing).@pl)

where, agt and obj refer agent and object relations respectively. The terms affect(icl>do), human(icl>animal) and environment(icl>abstract thing) are the UWs denoting concepts.

2.3 Attribute Labels (AL)

Speaker's view, aspect, time of event, etc. are captured by UNL attributes. For instance, in the above example, the attribute @entry denotes the main predicate of the sentence, @present denotes the present tense, @pl is the plural number and :01 is the scope ID.

UNL expressions provide the meaning of the text. Hence, search could be carried out considering the meaning rather than the text. This contributes to the development of a novel kind of search engine technology allowing information in one language can be stored in multiple languages. To convert Bangla

sentences into UNL form, we use EnConverter (EnCo) [1], a universal converter system provided by the UNL project shown in Figure 1. It is a language independent parser; a multi-headed Turing Machine [21] provides synchronously a framework for morphological, syntactic and semantic analysis. Natural language texts are analyzed sentence by sentence using a knowledge rich lexicon and by interpreting analysis rules. It scans an input string from left to right.

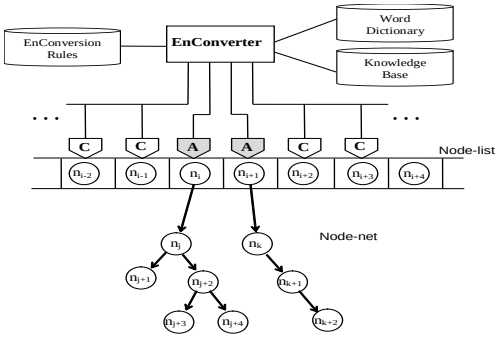

Figure 1. Structure of EnConverter ("A" indicates an Analysis Window, "C" indicates a Condition Window, and "nn" indicates an Analysis Node).

Moreover, when an input string is scanned all matched morphemes with the same string characters are retrieved from the dictionary and become the candidate morphemes according to the priority rule in order to build a syntactic tree and the semantic network for the sentence. The left character string is scanned from the beginning according to the applied rule. It moves back and forth over the Node List, which contains words of the input sentence. In the figure, "A", "C" and "n" indicate an Analysis Windows (AW), Condition Windows (CW) and "nn" indicates an Analysis Node respectively. The machine traverses the input sentence back and forth, retrieves the relevant dictionary entry from the Word Dictionary (Lexicon) depending on the attributes of the nodes under the AWs and those surrounding the CWs. It then generates the semantic relations between the UWs and /or attaches speech act attributes to them. As a result a set of UNL expressions is made equivalent of UNL graph [22]. EnCo is driven by a set of analysis rules to analyze a sentence using Word Dictionary and Knowledge Base. The enconversion rules have been described in [6]. Morphological analyses are performed by the left and right composition rules. This type of rule is used primarily for creating a syntactic tree with two nodes on the Analysis Windows. The semantic analyses are accomplished by either the left or right modification rules. They are used to create semantic relations between the words in a sentence [6].

3 Bangla Word Dictionary

The Word Dictionary is a collection of word dictionary entries. Each entry consists of three elements: Headword (HW), Universal Word (UW) and Grammatical Attribute (GA). A HW is a notation/surface of a native language word composing the input sentence. It is used as a trigger in obtaining equivalent UWs from a Word Dictionary in enconversion process. An UW expresses the meaning of a word which is used in creating UNL networks (i. e., UNL expressions) of output. GAs are the information on how words behave in a sentence and are used in enconversion rules. Each dictionary entry has the following format associating with any native language word [1, 6].

Data Format:

[HW]{ID}"UW"(Attribute1, Attribute2,...)<FLG, FRE, PRI>

Here,

HW ← Head Word (Bangla word)

ID ← Identification of Head Word (omissible)

UW ← Universal Word

ATTRIBUTE ← Attribute of the HW

FLG ← Language Flag

FRE ← Frequency of Head Word

PRI ← Priority of Head Word

Attributes denote the grammatical, semantic and morphological properties of a word. Some example entries of dictionary for Bangla language are given below:

[আপনি]{} "you(icl>person)" (PRON, HPRON, RES,SG,P2)

[ওরা]{} "they(icl>person)" (PRON, HPRON, PL,GEN, P3)

[আমি]{} "i(icl>person)" (PRON, HPRON, PL,RES, P3)

[তুই]{} "you(icl>person)" (SUB, PRON, HPRON, SG,NEG, P2)

where, PRON refers to Pronoun, HPRON to Human Pronoun, GEN to General, NEG to Neglect, RES to Respect, SUB for subject, SG to singular, PL to plural, CL to conversation language, LL to literature language, P1, P2, and P3 to first person, second and third persons respectively.

4 Analysis of Bangla Vowel Ended Roots and Verbal Inflexion

The root of a verb plays important role in forming verb of a sentence in any natural language. In order to analyze roots systematically, we have meticulously studied Bangla language grammars [12-17], verb and roots (vowel ended and consonant ended) [12-14] and morphological analysis [3, 18-21], based on their semantic structure. For an appropriate morphological analysis and designing verb root template, verb roots are divided, according to tenses and persons, into two broad categories: vowel ended group (VEG) and consonant ended group (CEG); each of them is then again divided into sub-groups. This paper focuses on only vowel ended groups. To date, 25 vowel ended roots have been identified in Bangla language [16, 20]. Through an extensive analysis of these roots we have categorized them into 11 subgroups: VEG1, VEG2, VEG3, VEG4, VEG5, VEG6, VEG7, VEG8, VEG9, VEG10 and VEG11 based on how verbal inflexions are added with them to form verbs. In categorization, the behavior of verbal inflexions with various kinds of persons (1st, 2nd and 3rd) and tenses (present, past and future) have been taken into consideration. For example: 'আমি বিশ্ববিদ্যালয়ে যাই', aami bishabiddaloye jai means "I go to university". Here, verb is 'যাই', jai. In this verb, root 'যা' is a vowel ended root (VER) and 'ই' is verbal inflexion (VI). If the above sentence is written in the present continuous form, it will be, 'আমি বিশ্ববিদ্যালয়ে যাচ্ছি', aami bishabiddaloye jachhi meaning "I am going to university". Although the root is same in both cases, the verbal inflexion for the later case is 'চ্ছি'. The present perfect form of this sentence is, 'আমি

বশি্ববদি্যালয়ে গয়িছে', Ammi bishabiddaloye giechhi meaning "I have gone to university". In this case, the original root 'যা' is changes its form to 'গি', gi' in generating verb 'গয়িছে', 'giechhi', where, 'য়ছে', 'echhi' is the verbal inflexion. Similar changes have been observed in different roots with different tenses. Bangla VERs have been further classified into the following three distinctive categorizes based on them.

4.1 Vowel Ended Roots and Their Verbal Inflexion for First Person

Tables 1, 2, and 3 present the subgroups of VEG1, VEG2, VEG3, VEG4, VEG5, VEG6, VEG7, VEG8, VEG9, VEG10 and VEG11 along with their alternatives and inflexions respectively. The tables show the roots with their corresponding tenses for first person as a subject.

In Table 1, roots পা (pa) and খা (kha) fall into VEG1. They do not change in present indefinite, present continuous, past continuous and future indefinite tenses. However, they are changed from পা (pa) to পে (pe) and খা (kha) to খে (khe) in other tenses. Similarly, roots গা (ga), চা (cha), and ছা (chha) in VEG2 are changed to গে (ge), চে (che), and ছে (chhe) in present perfect and past perfect tenses respectively. Roots নি (ni), and দি (di) of VEG3 remain unchanged in all tenses, whereas root যা (ja) in VEG4 is changed to গি (gi) in present perfect and past perfect tenses, গে (ge) in past indefinite and যে (je) in past habitual tenses respectively.

In Table 2, roots ছুঁ (cchu), থু (thu), শু (shu), ধু (dhu), ন (no), দু (du), নু (nu), রু (ru) and ল (lo) of VEG5, VEG6, VEG7 and VEG8 remain unchanged in all tenses. In Table 3, roots ধা (dha), না (na) and বা (ba) in VEG10 are changed into ধে (dhe), নে (ne) and বে (be) in present perfect and past perfect tenses and into ধাই (dhai), নাই (nai) and বাই (bai) in past indefinite and past habitual tenses respectively. And roots ক (ko), ব (bo), র (ro) and ল (lo) in VEG11 are changed to কই (koi), বই (boi), রই (roi) and লই (loi) in past indefinite and past habitual tenses respectively.

Table 1. Variation of Vowel Ended Roots and their Verbal Inflexions of VEG1 to VEG3 for First Person

Tenses	Vowel Ended Roots							
	পা (pa)	খা (kha)	গা (ga)	চা (cha)	ছা (ccha)	নি (ni)	দি (ni)	যা (ja)
Present Indefinite		ই	ই	ই	ই	ই	ই	ই
Present Continuous	চ্ছি	চ্ছি	চ্ছি	চ্ছি	চ্ছি	চ্ছি	চ্ছি	চ্ছি
Present Perfect	পা>পে য়ছে	খা>খে য়ছে	গা>গে য়ছে	চা>চে য়ছে	ছা>ছে য়ছে	য়ছে	য়ছে	যা>গি য়ছে
Past Indefinite	পা>পে লাম	খা>খে লাম	গা>গা ইলাম	চা>চা ইলাম	ছা>ছা ইলাম	লাম	লাম	যা>গে লাম
Past Habitual	পা>পে তাম	খা>খে তাম	গা>গাই তাম	চা>চাই তাম	ছা>ছাই তাম	তাম	তাম	যা>যে তাম
Past Continuous	চ্ছলিাম	চ্ছলিাম	চ্ছলিাম	চ্ছলিাম	চ্ছলিাম	চ্ছলিাম	চ্ছলিাম	চ্ছলিাম
Past Perfect	পা>পে য়ছেলিাম	খা>খে য়ছেলিাম	গা>গে য়ছেলিাম	চা>চে য়ছেলিাম	ছা>ছে য়ছেলিাম	য়ছেলিাম	য়ছেলিাম	যা>গি য়ছেলিাম
Future Indefinite	বে◌ো, ব	বে◌ো, ব	বে◌ো, ব	বে◌ো, ব	বে◌ো, ব	বে◌ো, ব	বে◌ো, ব	বে◌ো, ব
	VEG1		VEG2			VEG3		VEG4

Table 2. Variation of Vowel Ended Roots and their Verbal Inflexions of VEG5 to VEG8 for First Person

Tenses	Vowel Ended Roots								
	ছুঁ (chhu)	থু (thu)	শু (shu)	ধু (dhu)	ন (no)	দু (du)	নু (nu)	রু (ru)	লু (lo)
Present Indefinite	ই	ই	ই	ই	ই	ই	ই	ই	ই
Present Continuous	চ্ছি	চ্ছি	চ্ছি	চ্ছি		চ্ছি	চ্ছি	চ্ছি	চ্ছি
Present Perfect	য়ছেি	য়ছেি	য়ছেি	য়ছেি		য়ছেি	য়ছেি	য়ছেি	য়ছেি
Past Indefinite	লাম	লাম	লাম	লাম		লাম	লাম	লাম	লাম
Past Continuous	চ্ছিলাম	চ্ছিলাম	চ্ছিলাম	চ্ছিলাম		চ্ছিলাম	চ্ছিলাম	চ্ছিলাম	চ্ছিলাম
Past Perfect	য়ছেি লাম	য়ছেিলাম	য়ছেিলাম	য়ছেিলাম		য়ছেিলাম	য়ছেি লাম	য়ছেিলাম	য়ছেি লাম
Future Indefinite	ব	ব	ব	বো, ব		বো, ব	বো, ব	বো, ব	বো, ব
	VEG5				VEG6	VEG7			VEG8

Table 3. Variation of Vowel Ended Roots and their Verbal Inflexions of VEG9 to VEG11 for First Person

Tenses	Vowel Ended Roots						
	হ (ha)	ধা (dha)	না (na)	বা (ba)	ক (ko)	ব (bo)	র (ro)
Present Indefinite	ই	ই	ই	ই	ই	ই	ই
Present Continuous	চ্ছি	চ্ছি	চ্ছি	চ্ছি	চ্ছি	চ্ছি	চ্ছি
Present Perfect	য়ছেি	ধা>ধে য়ছেি	না>নে য়ছেি	বা>বে য়ছেি	য়ছেি	য়ছেি	য়ছেি
Past Indefinite	লাম	ধা>ধাই লাম	না>নাই লাম	বা>বাই লাম	ক>কই লাম	ব>বই লাম	র>লাম
Past Habitual	তাম	ধা>ধাই তাম	না>নাই তাম	বা>বাই তাম	ক>কই তাম	ব>বই তাম	র>রই তাম
Past Continuous	চ্ছিলাম	চ্ছিলাম	চ্ছিলাম	চ্ছিলাম	চ্ছিলাম	চ্ছিলাম	চ্ছিলাম
Past Perfect	য়ছেিলাম	ধা>ধে য়ছেিলাম	না>নে য়ছেিলাম	বা>বে য়ছেিলাম	য়ছেিলাম	য়ছেিলাম	য়ছেিলাম
Future Indefinite	ব	ব	ব	ব	বো, ব	বো, ব	বো, ব
	VEG9	VEG10			VEG11		

4.2 Vowel Ended Roots and Their Verbal Inflexion for Second Person

Tables 4-to-11 present the subgroups of VEG1-to-VEG10 along with their inflexions respectively. The tables show the roots with their corresponding tenses for second person as a subject. In Table 4, roots পা (pa) and খা (kha) in VEG1 are changed into পে (pe) and খে (khe) in present perfect, past indefinite, past habitual and past perfect tenses respectively. পা (pa) and খা (kha) are also changed into পে (pe) and খে (khe) for imperative in general (GEN) case. In Table 5, the roots গা (ga), চা (cha) and ছা (chha) in VEG2 are changed into গে (ge), চে (che) and ছে (chhe) in present perfect and past perfect tenses and into গাই (gai), চাই (chai) and ছাই (chhai) in past indefinite and past habitual tenses respectively. গা (ga), চা (cha) and ছা (chha) are also changed into গে (ge), চে (che) and ছে (chhe) in general form of second person for imperative tense.

Table 4. Variation of Vowel Ended Roots and their Verbal Inflexions of VEG1 for Second Person

	Vowel Ended Roots					
	পা (pa)			খা (kha)		
Tense	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)
Present Indefinite	ও	স	ন	ও	স	ন
Present Continuous	চ্ছ	চ্ছসি	চ্ছনে	চ্ছ	চ্ছসি	চ্ছনে
Present Perfect	পা>প য়ছে	পা>প য়ছেসি	পা>প য়ছেনে	খা>খ য়ছে	খা>খ য়ছেসি	খা>খ য়ছেনে
Imperative	ও	*	ন	ও	*	ন
Past Indefinite	পা>প লে	পা>প লি	পা>প লনে	খা>খ লে	খা>খ লি	খা>খ লনে
Past Habitual	পা>প তে	পা>প তি	পা>প তনে	খা>খ তে	খা>খ তি	খা>খ তনে
Past Continuous	চ্ছলে	চ্ছলি	চ্ছলিনে	চ্ছলে	চ্ছলি	চ্ছলিনে
Past Perfect	পা>প য়ছেলে	পা>প য়ছেলি	পা>প য়ছেলিনে	খা>খ য়ছেলে	খা>খ য়ছেলি	খা>খ য়ছেলিনে
Future Indefinite	ব	বি	বনে	ব	বি	বনে
Imperative	পা>পও	স	*	খা>খও	স	*
	VEG1					

Table 5. Variation of Vowel Ended Roots and their Verbal Inflexions of VEG2 for Second Person

	Vowel Ended Roots								
	গা (ga)			**চা** (cha)			**ছা** (chha)		
Tense	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)
Present Indefinite	ও	স	ন	ও	স	ন	ও	স	ন
Present Continuous	চ্ছ	চ্ছসি	চ্ছনে	চ্ছ	চ্ছসি	চ্ছনে	চ্ছ	চ্ছসি	চ্ছনে
Present Perfect	গা>গ য়ছে	গা>গ য়ছেসি	গা>গ য়ছেনে	চা>চ য়ছে	চা>চ য়ছেসি	চা>চ য়ছেনে	ছা>ছ য়ছে	ছা>ছ য়ছেসি	ছা>ছ য়ছেনে
Imperative	ও	*	ন	ও	*	ন	ও	*	ন
Past Indefinite	গা>গাইলে	গা>গাইলি	গা>গাইলনে	চা>চাইলে	চা>চাইলি	চা>চাইলনে	ছা>ছাইলে	ছা>ছাইলি	ছা>ছাইলনে
Past Habitual	গা>গাই তে	গা>গাই তি	গা>গাই তনে	চা>চাই তে	চা>চাই তি	চা>চাই তনে	ছা>ছাই তে	ছা>ছাই তি	ছা>ছাই তনে
Past Continuous	চ্ছলে	চ্ছলি	চ্ছলিনে	চ্ছলে	চ্ছলি	চ্ছলিনে	চ্ছলে	চ্ছলি	চ্ছলিনে
Past Perfect	গা>গ য়ছেলে	গা>গ য়ছেলি	গা>গ য়ছেলিনে	চা>চ য়ছেলে	চা>চ য়ছেলি	চা>চ য়ছেলিনে	ছা>ছ য়ছেলে	ছা>ছ য়ছেলি	ছা>ছ য়ছেলিনে
Future Indefinite	ব	বি	বনে	ব	বি	বনে	ব	বি	বনে
Imperative	গা>গ ও	স	*	চা>চ ও	স	*	ছা>ছ ও	স	*
	VEG2								

Table 6 shows the changes of root **নি** (ni) to **না** (na) and **নে**, root **দি** (di) to **দা** (da) and **দে** (de) in present indefinite, imparative and funture indefinite tenses and root **যা** (ja) to **গি** (gi) **গে** (ge) and **যে** (je) for present perfect, past indifinite, past habitual and past perfect recpectively. Roots **ছুঁ** (chhu), **থু** (thu), **শু** (shu) and **ধু** (dhu) are changed into **ছো** (chho), **থো** (tho), **শো** (sho) and **ধো** (dho) respectively in Table 7. In addition, Table 8 focuses the changes of roots **দু** (du) to **দো** (dho), **নু** (nu) to **নো** (no) and **রু** (ru) to **রো** (ro) in

present indefinite and imperative tenses and also roots **দু** (du) to **দুই** (dui), **নু** (nu) to **নই** (noi) and **রু** (ru) to **রুই** (rui) in past indefinite tenses respectively.

Table 6. Variation of Vowel Ended Roots and their Verbal Inflexions of VEG3 and VEG4 for Second Person

	Vowel Ended Roots								
	নি (ni)			**দি** (di)			**যা** (ja)		
Tense	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)
Present Indefinite	নচ্ছি ও	নচ্ছিস	নচ্ছিন	দচ্ছি ও	দচ্ছিস	দচ্ছিন	ও	যা	ন
Present Continuous	চ্ছ	চ্ছসি	চ্ছনে	চ্ছ	চ্ছসি	চ্ছনে	চ্ছ	চ্ছসি	চ্ছনে
Present Perfect	য়ছে	য়ছেসি	য়ছেনে	য়ছে	য়ছেসি	য়ছেনে	যা>গি য়ছে	যা>গি য়ছেসি	যা>গি য়ছেনে
Imperative	নচ্ছি ও	নচ্ছিস এ	নচ্ছিন	দচ্ছি ও	দচ্ছি এ	দচ্ছিন	ও	*	ন
Past Indefinite	লে	লি	লনে	লে	লি	লনে	যা>গে লে	যা>গে লি	যা>গে লনে
Past Habitual	তে	তি	তনে	তে	তি	তনে	যা>যে তে	যা>যে তি	যা>যে তনে
Past Continuous	চ্ছলিে	চ্ছলিি	চ্ছলিনে	চ্ছলিে	চ্ছলিি	চ্ছলিনে	চ্ছলিে	চ্ছলিি	চ্ছলিনে
Past Perfect	য়ছেলিে	য়ছেলিি	য়ছেলিনে	য়ছেলিে	য়ছেলিি	য়ছেলিনে	যা>গি য়ছেলিে	যা>গি য়ছেলিি	যা>গি য়ছেলিনে
Future Indefinite	নচ্ছি বে	বি	নচ্ছি বনে	দচ্ছি বে	বি	দচ্ছি বনে	বে	বি	বনে
Imperative	ও	স	*	ও	স	*	এও	স	*
	VEG3						**VEG4**		

Table 7. Variation of Vowel Ended Roots and their Verbal Inflexions of VEG5 for Second Person

	Vowel Ended Roots											
	ছুঁ (chhu)			**থু** (thu)			**শু** (shu)			**ধু** (dhu)		
Tense	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)
Present Indefinite	ছুঁ> ছ◌োও	ছুঁ> ছ◌োস	ছুঁ> ছ◌োন	থু> থ◌োও	থু> থ◌োস	থু> থ◌োন	শু> শ◌োও	শু> শ◌োস	শু> শ◌োন	ধু> ধ◌োও	ধু> ধ◌োস	ধু> ধ◌োন
Present Continuous	চ্ছ	চ্ছসি	চ্ছনে	চ্ছ	চ্ছসি	চ্ছনে	চ্ছ	চ্ছসি	চ্ছনে	চ্ছ	চ্ছসি	চ্ছনে
Present Perfect	য়ছে	য়ছেসি	য়ছেনে	য়ছে	য়ছেসি	য়ছেনে	য়ছে	য়ছেসি	য়ছেনে	য়ছে	য়ছেসি	য়ছেনে
Imperative	ও	*	ন	ও	*	ন	ও	*	ন	ও	*	ন
Past Indefinite	লে	লি	লনে	লে	লি	লনে	লে	লি	লনে	লে	লি	লনে
Past Habitua	তে	তি	তনে	তে	তি	তনে	তে	তি	তনে	তে	তি	তনে
Past Continuous	চ্ছলিে	চ্ছলিি	চ্ছলিনে	চ্ছলিে	চ্ছলিি	চ্ছলিনে	চ্ছলিে	চ্ছলিি	চ্ছলিনে	চ্ছলিে	চ্ছলিি	চ্ছলিনে
Past Perfect	য়ছেলিে	য়ছেলিি	য়ছেলিনে	য়ছেলিে	য়ছেলিি	য়ছেলিনে	য়ছেলিে	য়ছেলিি	য়ছেলিনে	য়ছেলিে	য়ছেলিি	য়ছেলিনে
Future Indefinite	বে	বি	বনে	বে	বি	বনে	বে	বি	বনে	বে	বি	বনে
Imperative	য়◌োও	স	*	য়◌োও	স	*	য়◌োও	স	*	য়◌োও	স	*
	VEG5											

Table 8. Variation of Vowel Ended Roots and their Verbal Inflexions of VEG6 and VEG7 for Second Person

	Vowel Ended Roots								
	দু (du)			নু (nu)			রু (ru)		
Tense	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)
Present Indefinite	দু>দে◌া ও	স	দু>দে◌া ন	নু>নে◌া ও	স	নু>নে◌া ন	রু>রে◌া ও	স	রু>রে◌া ন
Present Continuous	চ্ছ	চ্ছসি	চ্ছনে	চ্ছ	চ্ছসি	চ্ছনে	চ্ছ	চ্ছসি	চ্ছনে
Present Perfect	য়ছে	য়ছেসি	য়ছেনে	য়ছে	য়ছেসি	য়ছেনে	য়ছে	য়ছেসি	য়ছেনে
Imperative	দু>দে◌া ও	দু>দে◌া	ন	নু>নে◌া ও	নু>নে◌া	নু>নে◌া ন	রু>রে◌া ও	রু>রে◌া	রু>রে◌া ন
Past Indefinite	দু>দুই লে	দু>দুই র্লি	দু>দুই লনে	নু>নই লে	নু>র্লি	নু>নই লনে	রু>রুই লে	রু>রুই র্লি	রু>রুই লনে
Past Habitual	ইতে	ইর্তি	ইতনে	ইতে	ইর্তি	ইতনে	ইতে	ইর্তি	ইতনে
Past Continuous	চ্ছলিে	চ্ছর্লিি	চ্ছলিনে	চ্ছলিে	চ্ছর্লিি	চ্ছলিনে	চ্ছলিে	চ্ছর্লিি	চ্ছলিনে
Past Perfect	য়ছেলিে	য়ছেলিি	য়ছেলিনে	য়ছেলিে	য়ছেলিি	য়ছেলিনে	য়ছেলিে	য়ছেলিি	য়ছেলিনে
Future Indefinite	ইবে	ইবি	ইবনে	ইবে	ইবি	ইবনে	ইবে	ইবি	ইবনে
Imperative	ইও, ইয়ে◌া	ইস	ইবনে	ইও, ইয়ে◌া	ইস	ইবনে	ইও, ইয়ে◌া	ইস	ইবনে
	VEG6			VEG7					

Table 9. Variation of Vowel Ended Roots and their Verbal Inflexions of VEG8 and VEG9 for Second Person

	Vowel Ended Roots					
	ল (lo)			হ (ho)		
Tense	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)
Present Indefinite	ও	স	ন	ও	স	ন
Present Continuous	*	*	*	চ্ছ	চ্ছসি	চ্ছনে
Present Perfect	*	*	*	য়ছে	য়ছেসি	য়ছেনে
Imperative	ও		ন	ও	স	ওন
Past Indefinite	*	*	*	লে	র্লি	লনে
Past Habitual	*	*	*	তে	তসি	তনে
Past Continuous	*	*	*	চ্ছলিে	চ্ছর্লিি	চ্ছলিনে
Past Perfect	*	*	*	য়ছেলিে	য়ছেলিি	য়ছেলিনে
Future Indefinite	*	*	*	বে	বি	বনে
Imperative	ইও	ইস	ইবনে	ও	স	*
	VEG8			VEG9		

In Table 9, no changes have been made in roots since they can easily be combined with their inflexions in forming accurate verbs. Roots ধা (dha), না (na) and বা (ba) are changed into ধে (dhe), নে (ne) and বে (be) in present and past perfect tenses and the same roots are changing to ধাই (dhai), নাই (nai) and বাই (bai) for past indefinite and past habitual tenses respectively in Table 10. Changes also occure in imperative tense in the table. Table 11 demonstrates the verbal inflexions of roots ক (ko), ব (bo), র (ro) and স (so) for all forms of second person.

Table 10. Variation of Vowel Ended Roots and their Verbal Inflexions of VEG10 for Second Person

	Vowel Ended Roots								
	ধা (dha)			না (na)			বা (ba)		
Tense	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)
Present Indefinite	ও	স	ন	ও	স	ন	ও	স	ন
Present Continuous	চ্ছ	চ্ছসি	চ্ছনে	চ্ছ	চ্ছসি	চ্ছনে	চ্ছ	চ্ছসি	চ্ছনে
Present Perfect	ধা>ধে য়ছে	ধা>ধে য়ছেসি	ধা>ধে য়ছেনে	না>নে য়ছে	না>নে য়ছেসি	না>নে য়ছেনে	বা>বে য়ছে	বা>বে য়ছেসি	বা>বে য়ছেনে
Imperative	ও	*	ন	ও	*	ন	ও	*	ন
Past Indefinite	ধা>ধা ইলে	ধা>ধা লি	ধা>ধা ইলনে	না>না ইলে	না>না ইলি	না>নাই লনে	বা>বাই লে	বা>বাই লি	বা>বাই লনে
Past Habitual	ধা>ধাই তে	ধা>ধাই তি	ধা>ধাই তনে	না>নাই তে	না>নাই তি	না>নাই তনে	বা>বাই তে	বা>বাই তি	বা>বাই তনে
Past Continuous	চ্ছলে	চ্ছলি	চ্ছলিনে	চ্ছলে	চ্ছলি	চ্ছলিনে	চ্ছলে	চ্ছলি	চ্ছলিনে
Past Perfect	ধা>ধে য়ছেলে	ধা>ধে য়ছেলি	ধা>ধে য়ছেলিনে	না>নে য়ছেলে	না>নে য়ছেলি	না>নে য়ছেলিনে	বা>বে য়ছেলে	বা>বে য়ছেলি	বা>বে য়ছেলিনে
Future Indefinite	ইবে	ইবি	ইবনে	ইবে	ইবি	ইবনে	ইবে	ইবি	ইবনে
Imperative	ধা>ধে ও	স	*	না>নে ও	স	*	বা>বে ও	স	*
	Group VEG10								

Table 11. Variation of Vowel Ended Roots and their Verbal Inflexions of VEG11 for Second Person

	Vowel Ended Roots											
	ক (ko)			ব (bo)			র (ro)			স (so)		
Tense	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)	তুমি (Gen.)	তুই (Neg.)	আপনি (Res.)
Present Indefinite	ও	স	ন	ও	স	ন	ও	স	ন	ও	স	ন
Present Continuous	চ্ছ, ইছ	চ্ছসি, ইছসি	চ্ছনে , ইছনে	চ্ছ,ই ছ	চ্ছসি, ইছসি	চ্ছনে, ইছনে	চ্ছ,ই ছ	চ্ছসি, ইছসি	চ্ছনে, ইছনে	চ্ছ,ইছ	চ্ছসি, ইছসি	চ্ছনে , ইছনে
PresentPerfect	য়ছে	য়ছেসি	য়ছেন	য়ছে	য়ছেসি	য়ছেনে	য়ছে	য়ছেসি	য়ছেনে	য়ছে	য়ছেসি	য়ছেন
Imperative	ও	*	উন	ও	*	উন	ও	*	উন	ও	*	উন
PastIndefinite	ইলে	ইলি	ইলনে	ইলে	ইলি	ইলনে	ইলে	ইলি	ইলনে	ইলে	ইলি	ইলনে
Past Habitual	ইতে	ইতসি	ইতনে	ইতে	ইতসি	ইতনে	ইতে	ইতসি	ইতনে	ইতে	ইতসি	ইতনে
Past Continuous	চ্ছিলে	চ্ছলি	চ্ছিলনে	চ্ছিলে	চ্ছলি	চ্ছিলেন	চ্ছিলে	চ্ছলি	চ্ছিলেন	চ্ছলে	চ্ছলি	চ্ছিলনে
PastPerfect	য়ছেলে	য়ছেলি	য়ছেলনে	য়ছেলে	য়ছেলি	য়ছেলেন	য়ছেলে	য়ছেলি	য়ছেলেন	য়ছেলে	য়ছেলি	য়ছেলনে
Future Indefinite	বে	বি	বনে	বে	বি	বনে	বে	বি	বনে	বে	বি	বনে
Imperative	ইও	ইস	*	ইও	ইস	*	ইও	ইস	*	ইও	ইস	*
	Group VEG11											

4.3 Vowel Ended Roots and Their Verbal Inflexion for Third Person

Tables 12 to 17 present the subgroups of VEG1-to-VEG11 along with their alternatives and inflexions respectively. The tables show the roots with their corresponding tenses for third person as a subject.

Table 12. Variation of Vowel Ended Roots and their Verbal Inflexions of VEG1 and VEG2 for Third Person

Tense	Vowel Ended Roots									
	পা (pa)		খা (kah)		গা (ga)		চা (cha)		ছা (chha)	
	সে (Gen.)	তিনি (Res.)	সে (Gen.)	তিনি (Res.)	সে (Gen.)	তিনি (Res.)	সে (Gen.)	তিনি (Res.)	সে (Gen.)	তিনি (Res.)
Present Indefinite	য়	ন	য়	ন	য়	ন	য়	ন	য়	ন
Present Continuous	চ্ছে	চ্ছেন	চ্ছে	চ্ছেন	চ্ছে	চ্ছেন	চ্ছে	চ্ছেন	চ্ছে	চ্ছেন
Present Perfect	পা>পে য়ছে	পা>পে য়ছেন	খা>খে ছে	খা>খে ছেন	গা>গে য়ছে	গা>গে য়ছেন	চা>চে য়ছে	চা>চে য়ছেন	ছা>ছে য়ছে	ছা>ছে য়ছেন
Imperative	ক	ন	ক	ন	ক	ন	ক	ন	ক	ন
Past Indefinite	পা>পে ল	পা>পে লনে	খা>খে ল	খা>খে লনে	ইল	ইলনে	ইল	ইলনে	ইল	ইলনে
Past Habitual	পা>পে ত	পা>পে তনে	খা>খে ত	খা>খে তনে	ইত	ইতনে	ইত	ইতনে	ইত	ইতনে
Past Continuous	চ্ছিল	চ্ছিলেন	চ্ছিল	চ্ছিলেন	চ্ছিল	চ্ছিলেন	চ্ছিল	চ্ছিলেন	চ্ছিল	চ্ছিলেন
Past Perfect	পা>পে য়ছেলি	পা>পে য়ছেলেন	খা>খে য়ছেলি	খা>খে য়ছেলেন	গা>গে য়ছেলি	গা>গে য়ছেলেন	চা>চে য়ছেলি	চা>চে য়ছেলেন	ছা>ছে য়ছেলি	ছা>ছে য়ছেলেন
Future Indefinite	বে	বেন	বে	বেন	ইবে	ইবেন	ইবে	ইবেন	ইবে	ইবেন
Imperative	*	*	*	*	*	*	*	*	*	*
	Group VEG1				Group VEG2					

Table 13. Variation of Vowel Ended Roots and their Verbal Inflexions of VEG3 and VEG4 for Third Person

Tense	Vowel Ended Roots					
	নি (ni)		দি (di)		যা (ja)	
	সে (Gen.)	তিনি (Res.)	সে (Gen.)	তিনি (Res.)	সে (Gen.)	তিনি (Res.)
Present Indefinite	নি>নে য়	নি>নে ন	দি>দে য়	দি>দে ন	য়	ন
Present Continuous	চ্ছে	চ্ছেন	চ্ছে	চ্ছেন	চ্ছে	চ্ছেন
Present Perfect	য়ছে	য়ছেন	য়ছে	য়ছেন	যা>গি য়ছে	যা>গি য়ছেন
Imperative	ক	ন	ক	ন	ক	ন
Past Indefinite	ল	লনে	ল	লনে	যা>গে ল	যা>গে লনে
Past Habitual	ত	তনে	ত	তনে	যা>যে ত	যা>যে তনে
Past Continuous	চ্ছিল	চ্ছিলেন	চ্ছিল	চ্ছিলেন	চ্ছিল	চ্ছিলেন
Past Perfect	য়ছেলি	য়ছেলেন	য়ছেলি	য়ছেলেন	যা>গি য়ছেলি	যা>গি য়ছেলেন
Future Indefinite	নি>নে বে	নি>নে বেন	দি>দে বে	দি>দে বেন	বে	বেন
Imperative	*	*	*	*	*	*
	Group VEG3				Group VEG4	

Table 14. Variation of Vowel Ended Roots and their Verbal Inflexions of VEG5 for Third Person

Tense	Vowel Ended Roots							
	ছুঁ (chhu)		থু (thu)		শু (shu)		ধু (dhu)	
	সে (Gen.)	তিনি (Res.)	সে (Gen.)	তিনি (Res.)	সে (Gen.)	তিনি (Res.)	সে (Gen.)	তিনি (Res.)

Present Indefinite	ছুঁ>ছ্ঁো য়	ছুঁ>ছ্ঁো ন	থু>থ্ো য়	থু>থ্ো ন	শু>শ্ো য়	শু>শ্ো ন	ধু>ধ্ো য়	ধু>ধ্ো ন
Present Continuous	চ্ছে	চ্ছনে	চ্ছে	চ্ছনে	চ্ছে	চ্ছনে	চ্ছে	চ্ছনে
Present Perfect	য়ছে	য়ছেনে	য়ছে	য়ছেনে	য়ছে	য়ছেনে	য়ছে	য়ছেনে
Imperative	ক	ন	ক	ন	ক	ন	ক	ন
Past Indefinite	ল	লনে	ল	লনে	ল	লনে	ল	লনে
Past Habitual	ত	তনে	ত	তনে	ত	তনে	ত	তনে
Past Continuous	চ্ছলি	চ্ছলিনে	চ্ছলি	চ্ছলিনে	চ্ছলি	চ্ছলিনে	চ্ছলি	চ্ছলিনে
Past Perfect	য়ছেলি	য়ছেলিনে	য়ছেলি	য়ছেলিনে	য়ছেলি	য়ছেলিনে	য়ছেলি	য়ছেলিনে
Future Indefinite	বে	বনে	বে	বনে	বে	বনে	বে	বনে
Imperative	*	*	*	*	*	*	*	*
	Group VEG5							

Table 15. Variation of Vowel Ended Roots and their Verbal Inflexions of VEG6 and VEG7 for Third Person

	Vowel Ended Roots							
	ন (n)		**দু (du)**		**নু(nu)**		**রু (ru)**	
Tense	সে (Gen.)	তিনি (Res.)	সে (Gen.)	তিনি (Res.)	সে (Gen.)	তিনি (Res.)	সে (Gen.)	তিনি (Res.)
Present Indefinite	য়	ন	দু>দ্ো য়	দু>দ্ো ন	নু>ন্ো য়	নু>ন্ো ন	রু>র্ো য়	রু>র্ো ন
Present Continuous			চ্ছে	চ্ছনে	চ্ছে	চ্ছনে	চ্ছে	চ্ছনে
Present Perfect			য়ছে	য়ছেনে	য়ছে	য়ছেনে	য়ছে	য়ছেনে
Imperative			ক	ন	ক	ন	ক	ন
Past Indefinite			দু>দু ইল	দু>দুই লনে	নু>নুই ল	নু>নুই লনে	রু>রুই ল	রু>রুই লনে
Past Habitual			ইত	ইতনে	ইত	ইতনে	ইত	ইতনে
Past Continuous			চ্ছলি	চ্ছলিনে	চ্ছলি	চ্ছলিনে	চ্ছলি	চ্ছলিনে
Past Perfect			য়ছেলি	য়ছেলিনে	য়ছেলি	য়ছেলিনে	য়ছেলি	য়ছেলিনে
Future Indefinite			ইবে	ইবনে	ইবে	ইবনে	ইবে	ইবনে
Imperative			*	*	*	*	*	*
	Group VEG6		**Group VEG7**					

5 Formation of Template of Bangla Vowel Ended Roots

As per the detailed analyses of the Bangla VERs in above section, following template has been developed following the format defined in Section 3.

[HW]{}"UW(icl/iof…>concept1>concept2…,REL1>…,REL2>…," (ROOT, VEND, DEF/ ALT1/ ALT2/ALT3.., VEGn, #REL1, #REL2, …) <FLG, FRE, PRI>

where,

HW← Head Word (Bangla Word; in this case it is Bangla root);

UW← Universal Word (English word from knowledge base);

icl/iof/… means *inclusion/instance of …*to represent the concept of universal word

REL1/REL2.., indicates the related relations regarding the corresponding word.

ROOT ← it is an attribute for Bangla roots. This attribute is immutable for all Bangla roots.

Table 16. Variation of Vowel Ended Roots and their Verbal Inflexions of VEG8-to-VEG10 for Third Person

	Vowel Ended Roots									
	ল (lo)		**হ (ho)**		**ধা (dha)**		**না (na)**		**বা (ba)**	
Tense	সে	তিনি	সে	তিনি	সে	তিনি	সে	তিনি	সে	তিনি

	(Gen.)	(Res.)	(Gen.)	(Res.)	(Gen.)	(Res.)	(Gen.)	(Res.)	(Gen.)	(Res.)
Present Indefinite	য়	ন	য়	ন	য়	ন	য়	ন	য়	ন
Present Continuous	*	*	চ্ছে	চ্ছনে	চ্ছে	চ্ছনে	চ্ছে	চ্ছনে	চ্ছে	চ্ছনে
Present Perfect	*	*	য়ছে	য়ছেনে	ধা>ধে য়ছে	ধা>ধে য়ছেনে	না>নে য়ছে	না>নে য়ছেনে	বা>বে য়ছে	বা>বে য়ছেনে
Imperative	*	*	হ>হো ক	হ>হো ন	ক	ন	ক	ন	ক	ন
Past Indefinite	*	*	ল	লনে	ইল	ইলনে	ইল	ইলনে	ইল	ইলনে
Past Habitual	*	*	ত	তনে	ইত	ইতনে	ইত	ইতনে	ইত	ইতনে
Past Continuous	*	*	চ্ছলি	চ্ছলিনে	চ্ছলি	চ্ছলিনে	চ্ছলি	চ্ছলিনে	চ্ছলি	চ্ছলিনে
Past Perfect	*	*	য়ছেলি	য়ছেলিনে	ধা>ধে য়ছেলি	ধা>ধে য়ছেলিনে	না>নে য়ছেলি	না>নে য়ছেলিনে	বা>বে য়ছেলি	বা>বে য়ছেলিনে
Future Indefinite	*	*	বে	বনে	ইবে	ইবনে	ইবে	ইবনে	ইবে	ইবনে
Imperative	*	*	*	*	*	*	*	*	*	*
	Group VEG8		**Group VEG9**		**Group VEG10**					

Table 17. Variation of Vowel Ended Roots and their Verbal Inflexions of VEG11 for Third Person

	Vowel Ended Roots							
	ক (ko)		**ব (bo)**		**র (ro)**		**স (so)**	
Tense	সে (Gen.)	তিনি (Res.)	সে (Gen.)	তিনি (Res.)	সে (Gen.)	তিনি (Res.)	সে (Gen.)	তিনি (Res.)
Present Indefinite	য়	ন	য়	ন	য়	ন	য়	ন
Present Continuous	চ্ছে, ইছে	চ্ছনে, ইছনে	চ্ছে, ইছে	চ্ছনে, ইছনে	চ্ছে, ইছে	চ্ছনে, ইছনে	চ্ছে, ইছে	চ্ছনে, ইছনে
Present Perfect	য়ছে	য়ছেনে	য়ছে	য়ছেনে	য়ছে	য়ছেনে	য়ছে	য়ছেনে
Imperative	উক	উন	উক	উন	উক	উন	উক	উন
Past Indefinite	ইল	ইলনে	ইল	ইলনে	ইল	ইলনে	ইল	ইলনে
Past Habitual	ত	তনে	ত	তনে	ত	তনে	ত	তনে
Past Continuous	চ্ছলি	চ্ছলিনে	চ্ছলি	চ্ছলিনে	চ্ছলি	চ্ছলিনে	চ্ছলি	চ্ছলিনে
Past Perfect	য়ছেলি	য়ছেলিনে	য়ছেলি	য়ছেলিনে	য়ছেলি	য়ছেলিনে	য়ছেলি	য়ছেলিনে
Future Indefinite	ইবে	ইবনে	ইবে	ইবনে	ইবে	ইবনে	ইবে	ইবনে
Imperative	*		*	*	*	*	*	*
	Group VEG11							

VEND is the attributes for vowel ended roots.

VEGn ← attribute for the group number of vowel ended roots (n=1, 2…10).

DEF/ALT1/ALT2/ALT3 etc. are the attributes for the default, first, second or third alternatives of the vowel ended roots respectively.

#REF1, #REF2 etc. are the possible corresponding relations regarding the root word.

In our dictionary we avoid <FLG, FRE, PRE> part of the entry as it is mostly same for all entries.

Here, attributes, ROOT and VEND are fixed for all Bangla vowel ended roots, whereas ALT1, ALT2 or ALT3 etc. are not necessary for all roots, they are used only for alternative roots.

In the following examples we construct the dictionary entries for some sample verb roots using our designed template:

[যা]{}"go(icl>move>do, plf>place, plt>place, agt>thing)" (ROOT, VEND, VEG3, #PLF, #PLT, #AGT)

[গি]{}"go(icl>move>do, plf>place, plt>place, agt>thing)" (ROOT, VEND, ALT1, VEG3, #PLF, #PLT, #AGT)

[খা]{}"eat(icl>consume>do,agt>living_thing, ins>thing, obj>concrete_thing, plf>thing, tim>abstract_thing)" (ROOT, VEND, VEG1, #PLF, #PLT, #AGT)

For first two entries the relation *plf* (place from) indicates from where agent go/goes, plt (place to) means to where go/goes, agt (agent) for who go/goes and attribute ALT1 indicates that root 'গি' (*gi*) is the first alternative of root 'যা' (*ja*) shown in Table 1. Attributes #PLF, #PLT and #AGT indicate that relations *plf, plt* and *agt* can be made with roots 'গি' (*gi*) and 'যা' (*ja*). Similarly, other entries have been developed according to the format discussed above. Our proposed dictionary entries of VERs along with their alternatives are given below.

- **Dictionary Entries of VEG1:**

[পা]{}"get((icl>do,equ>obtain,src>uw,agt>thing,obj>thing)" (ROOT, VEND, DEF, VEG1, #OBJ,#AGT)

[পে]{}"get((icl>do,equ>obtain,src>uw,agt>thing,obj>thing)" (ROOT, VEND, ALT1, VEG1, #OBJ,#AGT)

[খা]{}"eat(icl>consume>do,agt>living_thing,obj>concrete_thing,ins>thing)"(ROOT,VEND, DEF,VEG1,#AGT, #OBJ,#INS)

[খে]{}"eat(icl>consume>do,agt>living_thing,obj>concrete_thing,ins>thing)"(ROOT,VEND, ALT1,VEG1,#AGT, #OBJ,#INS)

- **Dictionary Entries of VEG2:**

[গা]{}"sing(icl>do,com>music,cob>thing,agt>living_thing,obj>song,rec>living_thing)" (ROOT, VEND, DEF, VEG2,#AGT,#OBJ,#COM,#COB,#REC)

[গে]{}"sing(icl>do,com>music,cob>thing,agt>living_thing,obj>song,rec>living_thing)" (ROOT, VEND, ALT1, VEG2, #AGT,#OBJ,#COM,#COB,#REC)

[গাই]{}"sing(icl>do,com>music,cob>thing,agt>living_thing,obj>song,rec>living_thing)" (ROOT, VEND, ALT2, VEG2, #AGT,#OBJ,#COM,#COB,#REC)

[চা]{}"want(icl>desire>be,obj>uw,aoj>volitional_thing,pur>thing)"(ROOT,VEND,DEF,VEG2,#OBJ,#AOJ, #PUR)

[চে]{}"want(icl>desire>be,obj>uw,aoj>volitional_thing,pur>thing)"(ROOT,VEND,ALT1,VEG2,#OBJ,#AOJ, #PUR)

[চাই]{}"want(icl>desire>be,obj>uw,aoj>volitional_thing,pur>thing)"(ROOT,VEND,ALT2,VEG2,#OBJ,#AOJ, #PUR)

[ছা]{}"roof(icl>cover>do,agt>volitional_thing,obj>thing,ins>thing)"(ROOT,VEND,DEF,VEG2,#AGT,#OBJ,#INS)

[ছে]{}"roof(icl>cover>do,agt>volitional_thing,obj>thing,ins>thing)"(ROOT,VEND,ALT1,VEG2,#AGT,#OBJ,#INS)

[ছাই]{}"roof(icl>cover>do,agt>volitional_thing,obj>thing,ins>thing)"(ROOT,VEND,ALT2,VEG2,AGT,#OBJ,#INS)

- **Dictionary Entries of VEG3:**

[নি]{}"take(icl>capture>do,agt>thing,obj>thing)"(ROOT, VEND, DEF,VEG3, #AGT, #OBJ)<B,0,0>

[দি]{}"give(icl>do,equ>hand_over,agt>living_thing,obj>concrete_thing,rec>person)" (ROOT, VEND, DEF, VEG3, #AGT,#OBJ,#REC)

- **Dictionary Entries of VEG4:**

[যা]{}"go(icl>move>do, plf>place, plt>place, agt>thing)" (ROOT, VEND, DEF, VEG4, #PLF, #PLT, #AGT)

[গি]{}"go(icl>move>do, plf>place, plt>place, agt>thing)" (ROOT, VEND, ALT1, VEG4, #PLF, #PLT, #AGT)

[গে]{}"go(icl>move>do, plf>place, plt>place, agt>thing)" (ROOT, VEND, ALT2, VEG4, #PLF, #PLT, #AGT)

[যে]{}"go(icl>move>do, plf>place, plt>place, agt>thing)" (ROOT, VEND, ALT3, VEG4, #PLF, #PLT, #AGT)

- **Dictionary Entries of VEG5:**

[ছুঁ]{}"touch(icl>come_in_contact>do,agt>person,obj>concrete_thing,ins>thing"(ROOT,VEND,DEF,VEG5, #AGT, #OBJ,#INS)

[থু]{}"put(icl>displace>do,plc>thing,agt>thing,obj>thing)"(ROOT,VEND,DEF, VEG5, #AGT, #OBJ,#PLC)

[শু]{}"sleep(icl>rest>be,aoj>living_thing)"(ROOT,VEND,VEG5,#AOJ,#PLC)<B,0,0>

[ধু]{}"wash(icl>serve>do,agt>living_thing,obj>concrete_thing,ins>functional_thing)"(ROOT,VEND,DEF, VEG5, #AGT,#OBJ,#INS)

- Dictionary Entries of VEG6:

[না]{}"be(icl>be>not, aoj>thing)" (ROOT, VEND, DEF, VEG6, #AOJ)

- **Dictionary Entries of VEG7:**

[দু]{}"milk(icl>draw>do,agt>thing,obj>thing)" (ROOT, VEND, DEF, VEG7, #AGT, #OBJ)

[নু]{}"bath(icl>vessel>thing)" (ROOT, VEND, VEG7, #PLF, #PLT, #AGT)

[রু]{}"sow(icl>put>do,plt>thing,agt>thing,obj>concrete_thing)"(ROOT,VEND,DEF,VEG7,#PLT, #AGT,#OBJ)

- **Dictionary Entries of VEG8:**

[ল]{}"take(icl>require>be,obj>thing,aoj>thing,ben>person)" (ROOT, VEND, DEF, VEG8, #OBJ, #AOJ, #BEN)

- **Dictionary Entries of VEG9:**

[হ]{}"be(icl>be,equ>be_located,aoj>thing,plc>uw)"(ROOT,VEND,DEF,VEG9,#AOJ, #PLC)

- **Dictionary Entries of VEG10:**

[ধা]{}"urge(icl>rede>do,agt>volitional_thing,obj>volitional_thing,gol>thing)"(ROOT,VEND,DEF,VEG10,#AGT, #OBJ,#GOL)

[না]{}"bath(icl>vessel>thing)" (ROOT, VEND, VEG10,#AGT,#PLC)

[বা]{}"row(icl>move(icl>cause)>do,plt>thing,agt>person,obj>boat,ins>thing)"(ROOT,VEND,DEF,VEG10,#PLF, #PLT, #AGT,#OBJ,#INS)

- **Dictionary Entries of VEG11:**

[ক]{}"talk(icl>communicate>do,cob>uw,agt>person,obj>thing,ptn>person)"(ROOT,VEND,DEF,VEG11,#AGT, #OBJ,#PTN,#COB)

[ব]{}"bear(icl>have>be,obj>property,aoj>thing)"(ROOT,VEND,DEF, VEG11, #OBJ, #AOJ)

[র]{}"stay(icl>dwell>be,aoj>person,plc>uw)" (ROOT, VEND, DEF, VEG11, #AOJ, #PLC)

6 Formation of Template for Verbal Inflexion

In the previous section, we outlined a template for Bangla verb roots. However, the template for verbal inflexion (VI) is very similar to that of Bangla verb roots with only a difference is that the later one does not have any universal word and that differs from the former with attributes they use. Template of Verbal Inflexions is as follows:

[HW]{} "" (VI, V, Pn [,ALT1/ALT2,ALT3...], GEN/RES/NEG, Atense, LL/CL, VEG_n/ ^VEG_n) <FLG, FRE, PRI>

HW← Head Word (Verbal Inflexion of Bangla Verb Root); UW← Universal Word (In case of Verbal Inflexion, UW is null); VI← is an attribute of Verbal Inflexion, V← for Verb, since Verbal Inflexions form verb when added with Bangla verb root as Suffixes, so the 'V' is considered as an attribute.

Pn (n=1 to 3) ← Attribute for person; P1, P2 and P3 refer first, second and third persons respectively. These are important attributes because verb varies according to persons.

ALT1/ALT2/ALT3 ← Attributes for alternative roots. These attributes are used as attributes of verbal inflexions when they are combined with the respective verb roots.

GEN/RES/NEG← Attributes for verbal inflexions when they are combined with verb roots to form general (GEN), respective (RES) and neglect (NEG) verbs in respect to person. They are used as attributes with the VIs that are combined with verb roots to form verb only for second and third persons.

Atense ← Attribute Tense; - this is also an important attribute because verb varies according to Bangla Tenses.

LL/CL← Attribute for types of languages where LL refers to literature language and CL to conversation language. They are used as attributes with the VIs as they form LL or CL types of verbs.

VEG_n//^VEG_n← Attributes indicate for vowel ended group number or not for vowel ended group. They are used as attributes of VIs as they are combined with respective groups or not. Similar to *verb roots* attribute, VI is fixed for all *Verbal Inflexions.* Attribute Pn can be either attributes 'P1' (for first person), 'P2' (for second person) or 'P3' (for third person). Again Atense can be any tense such as attributes 'PRS' (for present indefinite), 'PRG' (progress for present continuous) 'CMPL' (complete for perfect tense), 'IMP' for imperative and 'HAB' for habitual etc. If the tense is past continuous, two attributes are used consecutively such as attribute 'PST' (for past) and 'PRG' (for continuous) and 'FUT' for future tense.

Some examples of dictionary entries of *Verbal Inflexions* according to the proposed template are given below:

[য়ছেলিাম] " "{}(VI,P1,PST, PER,ALT1,CH,VEG1,VEG2, VEG9)

[চ্ছলিাম] " "{}(VI,P1,PST,PRG,CH)

[বা] " "{}(VI,P2,NEG,FUT,CH)

[চ্ছনে] " "{}(VI,P2,RES,PRT, PRG,CH)

Here, VI, 'য়ছেলিাম' can be combined with first alternative roots (as attribute ALT1 is used to define first alternative root) with verb roots of *vowel ended group 1* or *vowel ended group 2* for past perfect tense (attributes PST for past and CMPL for perfect) to create the verbs of conversation language (CL attribute for conversation language) for first person (attribute is P1). Similarly, attributes for other dictionary entries are defined. Our proposed dictionary entries of verbal inflexions are as follows.

- **Dictionary entries of verbal inflexions of all tenses for first person as a subject:**
  [ই] " "{}(VI, 1P, PRS, DEF, CL)

  [চ্ছি] " "{}(VI, 1P, PRS, PRG, DEF, CL)

  [য়ছেি] " "{}(VI, 1P, PRS, CMPL, DEF, ALT1, CL, VEG1, VEG2, VEG4,VEG10)

  [লাম] " "{}(VI,1P,PST,ALT1, ALT2,CL, VEG1, VEG2, VEG4,VEG10)

  [তাম] " "{}(VI,P1,PST,DEF,ALT1,ALT2,ALT3, SHD,VEG3)

  [চ্ছলিাম] " "{}(VI,1P,PST, PRG, DEF,CL)

  [য়ছেলিাম] " "{}(VI,1P,PST, CMPL,ALT1,CL,VEG1,VEG2, VEG4,VEG10)

  [ব] " "{}(VI, 1P, FUT, DEF, CL)

  [বে◌া] " "{}(VI, 1P, FUT, DEF, CL)

- **Dictionary entries of verbal inflexions of all tenses for second person as a subject:**
  [ও] " "{}(VI, 2P, PRS, DEF, CL,DEF,ALT1,VEG3,VEG5,VEG7,GEN)

[স্] " "{}(VI, 2P, PRS, PRG, CL,DEF,ALT1,VEG3,VEG5,VEG7,NEG)

[ন] " "{}(VI, 2P, PRS, CMPL, DEF, ALT1, CL, VEG3,VEG5,VEG7,RES)

[চ্ছ] " "{}(VI, 2P, PRS, PRG, DEF, CL,GEN)

[চ্ছসি] " "{} (VI, 2P, PRS, PRG, DEF, CL,NEG)

[চ্ছনে] " "{} (VI, 2P, PRS, PRG, DEF, CL,RES)

[য়ছে] " "{}(VI, 2P, PRS, CMPL, DEF, ALT1,VEG1, VEG2, VEG4, VEG10, CL,GEN)

[য়ছেসি] " "{} (VI, 2P, PRS, CMPL, DEF, ALT1,VEG1, VEG2, VEG4, VEG10, CL,NEG)

[য়ছেনে] " "{} (VI, 2P, PRS, CMPL, DEF, ALT1,VEG1, VEG2, VEG4, VEG10, CL,RES)

[লে] " "{}(VI, 2P, PST, DEF, ALT1,ALT2,VEG1, VEG2, VEG4, VEG7, CL,GEN)

[লি] " "{} (VI, 2P, PST, DEF, ALT1,ALT2,VEG1, VEG2, VEG4, VEG7, CL,NEG)

[লনে] " "{} (VI, 2P, PST, DEF, ALT1,ALT2,VEG1, VEG2, VEG4, VEG7, CL,RES)

[তে] " "{}(VI, 2P, PST, HAB, DEF, ALT1,ALT2,VEG1, VEG1, VEG2, VEG4, VEG10, CL,GEN)

[তি] " "{} (VI, 2P, PST, HAB, DEF, ALT1,ALT2,VEG1, VEG1, VEG2, VEG4, VEG10, CL,NEG)

[তনে] " "{} (VI, 2P, PST, HAB, DEF, ALT1,ALT2,VEG1, VEG1, VEG2, VEG4, VEG10, CL,RES)

[চ্ছলিে] " "{}(VI, 2P, PST, PRG, DEF,CL,GEN)

[চ্ছলিি] " "{} (VI, 2P, PST, PRG, DEF,CL,GEN)

[চ্ছলিনে] " "{} (VI, 2P, PST, PRG, DEF,CL,GEN)

[য়ছেলিে] " "{}(VI, 2P, PST, CMPL, DEF, ALT1, VEG1, VEG2, VEG4, CL,GEN)

[য়ছেলি ি] " "{} (VI, 2P, PST, CMPL, DEF, ALT1, VEG1, VEG2, VEG4, CL,NEG)

[য়ছেলিনে] " "{} (VI, 2P, PST, CMPL, DEF, ALT1, VEG1, VEG2, VEG4, CL,RES)

[বে] " "{}(VI, 2P, FUT, DEF, ALT1, ALT2, VEG3, VEG4, CL,GEN)

[বি] " "{} (VI, 2P, FUT, DEF, ALT1, ALT2, VEG3, VEG4, CL,NEG)

[বনে] " "{} (VI, 2P, FUT, DEF, ALT1, ALT2, VEG3, VEG4, CL,RES)

[ইবে] " "{} (VI, 2P, FUT,DEF, CL, VEG7, VEG10,GEN)

[ইবি] " "{} (VI, 2P, FUT,DEF, CL, VEG7, VEG10,NEG)

[ইবনে] " "{}(VI, 2P, FUT,DEF, CL, VEG7, VEG10,RES)

[ও] " "{}(VI, 2P, IMPR, ALT1,VEG3,VEG5,VEG7,GEN)

[য়ো] " "{}(VI, 2P, IMPR, CL,DEF,VEG5)

[ইও] " "{}(VI, 2P, IMPR, DEF, CL,VEG5,GEN)

[ইস] " "{} (VI, 2P, IMPR, CMPL, DEF, CL,VEG5,NEG)

[ইবনে] " "{} (VI, 2P, IMPR, DEF, CL,VEG5,RES)

- **Dictionary entries of verbal inflexions of all tenses for third person as a subject:**

[য়] " "{} (VI, 3P, PRS, DEF, ALT1, CL,VEG3, VEG5, VEG7, GEN)

[ন] " "{} (VI, 3P, PRS, DEF, ALT1, CL,VEG3, VEG5, VEG7, RES)

[চ্ছে] " "{} (VI, 3P, PRS, PRG, DEF, CL, GEN)

[চ্ছনে] " "{} (VI, 3P, PRS, PRG, DEF, CL, RES)

[য়ছে] " "{} (VI, 3P, PRS, CMPL, DEF, ALT1,CL,VEG1, VEG2, VEG4, VEG10,GEN)

[য়ছেনে] " "{} (VI, 3P, PRS, CMPL, DEF, ALT1,CL,VEG1, VEG2, VEG4, VEG10,RES)

[ক] " "{} (VI, 3P, IMP, DEF, ALT1,CL, VEG9,GEN)

[উক] " "{} (VI, 3P, IMP, DEF,CL,VEG11,GEN)

[উন] " "{} (VI, 3P, IMP, DEF, CL,VEG11, RES)

[ল] " "{} (VI, 3P, PST, DEF, ALT1, ALT2, CL,VEG1, VEG2, VEG4,GEN)

[লনে] " "{} (VI, 3P, PST, DEF, ALT1, ALT2, CL,VEG1, VEG2, VEG4, RES)

[ইল] " "{} (VI, 3P, PST, DEF, CL, VEG2, VEG10, VEG11, GEN)

[ইলনে] " "{} (VI, 3P, PST, DEF, CL, VEG2, VEG10, VEG11, RES)

[ত] " "{} (VI, 3P, PST, HAB, DEF, ALT1, ALT2, CL,VEG1, VEG4,GEN)

[তনে] " "{} (VI, 3P, PST, HAB, DEF, ALT1, ALT2, CL,VEG1, VEG4, RES)

[ইত] " "{} (VI, 3P, PST, HAB, DEF, CL,VEG2, VEG7,VEG10, GEN)

[ইতনে] " "{} (VI, 3P, PST, HAB, DEF, CL,VEG2, VEG7,VEG10, RES)

[চ্ছিল] " "{} (VI, 3P, PST, PRG, DEF, CL, GEN)

[চ্ছিলিনে] " "{} (VI, 3P, PST, PRG, DEF, CL, RES)

[য়ছেলি] " "{} (VI, 3P, PST, CMPL, DEF, ALT1, CL, VEG1, VEG2, VEG4,VEG10, GEN)

[য়ছেলিনে] " "{} (VI, 3P, PST, CMPL, DEF, ALT1, CL, VEG1, VEG2, VEG4,VEG10, RES)

[বে] " "{} (VI, 3P,FUT, DEF, ALT1, CL, VEG3, GEN)

[বনে] “ ”{} (VI, 3P,FUT, DEF, ALT1, CL, VEG3, RES)

[ইবে] “ ”{} (VI, 3P, FUT, DEF, CL, VEG2, VEG7,VEG10, VEG11, GEN)

[ইবনে] “ ”{} (VI, 3P, FUT, DEF, CL, VEG2, VEG7,VEG10, VEG11, RES)

7 Conversion of a Bangla Sentence into UNL Expression

The encoding process is performed by shift/reduce parsing [22-23]. To explain the encoding steps, we give an example of a simple Bangla assertive sentence. Assertive simple sentences have only one main clause. We assume that analysis rules and the dictionary of Bangla to UNL are given to the analyser system *EnCo*. The following Bangla sentence is considered as an example

Bangla sentence: আমরা আম খাইতছে।

:Transliterated sentenceAmra aam khaitechi·

·Equivalent English sentence: We are eating mango

] sentence is processed according to the algorithm that we have developed in The input Bangla24]The .
.chunks obtained from the input sentence are given below

(আমরা) (আম) (খা) (ইতছে)

(Amra) (aam) (kha) (itechi)

We have used an EnConverter [25] tool for our experiment. The tool takes a dictionary file for the sentence shown in Table 18 and a set of analysis rules shown in Table 19 as its input.

Table 18. Dictionary entries of respective Bangla sentence

[আমরা]{} “we(icl>group)”(PRON, HPRON, P1, PL, SUBJ)
[আম]{}“mango(icl>edible_fruit>thing)”(N, NCOM, FRUIT)
[খা]{}“eat(icl>consume>do,agt>living_thing,obj>concrete_thing)”(ROOT,VEND,VEG1, #AGT, OBJ)
[ইতছে]{}”INF” (VI,VEND,P1,PRS,PRG)

In Table 18, attributes PRON indicates pronoun, HPRON indicates human pronoun, P1 for first person, PL for plural, SUBJ for subject, N indicates noun, NCOM for common noun, FRUIT for fruit item, ROOT for verb root, VEND for vowel ended root, PRS for present tense, PRG for progress means present continuous rense respectively.

EnCo can input either a string or a list of words for a sentence of a native language. A list of morphemes or words of a sentence must be enclosed by [<<] and [>>] [1]. When the sentence is taken into EnCo, it places the sentence head (<<) in the LAW (Left Analysis Window), sentence texts or morphemes or words in the RAW (Right Analysis Window) and the sentence tail (>>) in the RCW (Right Condition Window) shown in Figure 2. After insertion of the input file with our given sentence the rules shown in Table 19 will be applied step by step to complete the conversion processes of the sentence to UNL expressions. Rule 1 describes when sentence head is in the LAW and subject ‘আমরা’ *aamra* (we) is in the RAW then AWs will be shifted to right after rule application. The EnCo will then retrieve the word, ‘আমরা’ from the Word Dictionary file and remains in the LAW and ‘আম খাইতছে’, *aam khaitechi* (mango eating) will be in

the RAW. Rule 2 is applied to delete the right node which is a blank space between 'আমরা' and noun 'আম', *aam* (mango) and only the noun 'আম' will be placed in the RAW, while the verb 'খাইতেছি', *khaitechi* (eating) will be placed in the RCW. Rule 3 is then applied to shift the windows to right and Rule 4 is applied to delete the space between 'আম' (aam) and 'খাইতেছি' (khaitechi) so that the word 'আম' (aam) is retrieved from the Word Dictionary and remains in the LAW and the verb 'খাইতেছি' (khatechi) is divided into root 'খা' (kha) which remains in the RAW and verbal inflexion 'ইতেছি' (itechi) remains in the RCW. To perform morphological analysis Rule 5 is now applied to place root 'খা' (kha) in the LAW and verbal inflexion 'ইতেছি' (itechi) in the RAW. At this time, Enco retrieves the dictionary entries 'খা' (kha) and 'ইতেছি' (itechi) from the word dictionary (input file) and will apply Rule 6 to combine the nodes of left and right analysis windows into a composite node to complete the morphological analysis of the verb 'খাইতেছি' (khaitechi). Then Rule 7 rewrites the attributes by deleting VI, VEND, and CEND for verb 'খাইতেছি' (khaitechi) that remains in the RAW.

After completion of the morphological analysis, Rule 8 is applied to perform semantic analysis between noun 'আম' (aam) and verb 'খাইতেছি' (khaitechi) by object relation, *obj* and noun 'আম' (aam) is deleted from the node-list, where 'খাইতেছি'(khaitechi) remains in the RAW. Similarly, another semantic analysis is held by agent relation, *agt* between the subject 'আমরা' (aamra) and verb 'খাইতেছি' (khaitechi) after applying Rule 9. The word 'আমরা' (aamra) is deleted from the node-list and the verb 'খাইতেছি' (khaitechi) remains in the RAW, which is the main predicate of the sentence. Later Rule 10 is applied to shift the windows to right and *&@entry* attribute is added to the verb as verb 'খাইতেছি' (khaitechi) is the main word of the sentence.

Finally, Rule 11 is applied to place the sentence tail (STAIL) on the LAW to complete the conversion process. After completion of the conversion process, the following UNL expresson will be created by the EnConverter shown in Table 20.

Table 19. Dictionary entries of respective Bangla sentence

Rule	Description
Rule 1: R{SHEAD:::}{PRON,SUBJ:::}P10;	Right Shift Rule
Rule 2: DR{SUBJ,^blk:blk::}{BLK:::}P10;	Right Deletion Rule
Rule 3: R{PRON,SUBJ:::}{N:::}P10;	Right Shift Rule
Rule 4: DR{N,^blk:blk::}{BLK:::}P10;	Right Deletion Rule
Rule 5: R{N:::}{ROOT,^VERB:::}P10;	Right Shift Rule
Rule 6: +{ROOT,VEND,^ALT,^VERB:+ VERB,- ROOT, +@::} {VI,VEND:::}P10;	Left Composition Rule
Rule 7: :{:::}{VERB,VI:-KBIV,-VEND,-CEND::}P10;	Insertion Rule
Rule 8: >{N::obj:}{VERB,#OBJ:::}P10;	Right Modification Rule
Rule 9: >{HPRON,SUBJ::agt:}{VERB,#AGT:::}P10;	Right Modification Rule
Rule 10: R{SHEAD:::}{VERB,^&@entry:+&@entry::}P10;	Right Shift Rule
Rule 11: R{VERB:::}{STAIL:::}P10;	Right Shift Rule

Table 20. UNL expression of the converted sentence

{org:en}

```
We are eating mango.
{/org}
{unl}
agt(eat(icl>consume>do,agt>living_thing,obj>concrete_thing,ins>thing)
.@entry.@pl.@present.@progress,we(icl>group).@pl)
obj(eat(icl>consume>do,agt>living_thing,obj>concrete_thing,ins>thing)
.@entry.@pl.@present.@progress,mango(icl>edible_fruit>thing))
{/unl}
```

Figure 2. Initial state of the Analysis Windows and the node list

8 Conclusions and Future Works

This paper has explored the Bangla vowel ended roots and grouped them into different categories based on how verbal inflexions are added with them to from verbs for all persons and tenses. This paper has also outlined the formats of word dictionary for the vowel ended roots and verbal inflexions, and developed required dictionary entries related to them. These entries can be used to generate verbs combining with their respective verbal inflexions. Our experimental result shows that Bangla native language sentences with verb can now be easily converted into UNL expression by analysis rules. The proposed format can be equally applicable to other languages with vowel ended roots. Our future research is to develop formats for Bangla consonant ended roots for first, second and third persons in all tenses.

REFERENCES

[1] Uchida, H., Zhu, M., and Senta, T. C. D., *Universal Networking Language*, UNDL Foundation, International environment house, 2005/6, Geneva, Switzerland.

[2] Choudhury, M. E. H., Ali, M. N.Y., Sarkar, M.Z.H., and Ahsan, R., *Bridging Bangla to Universal Networking Language- a Human Language Neutral Meta- Language*, International Conference on Computer and Information Technology (ICCIT), Dhaka, 2005, pp. 104-109.

[3] Ali, M.N.Y., Das, J.K., Mamun, S.M. A. A., and Nurannabi, A.M., *Morphological Analysis of Bangla Words for Universal Networking Language*, Third International Conference on Digital Information Management (ICDIM 2008), London, England. pp. 532-537.

[4] Ali, M. N. Y., Das, J. K., Al Mamun, S. M. A., and Choudhury, M. E.H., *Specific Features of a Converter of Web Documents from Bengali to Universal Networking Language*, International Conference on Computer and Communication Engineering 2008 (ICCCE'08), Kuala Lumpur, Malaysia, pp. 726-731.

[5] Serraset, G. and Boitet, C., *UNL French Deconversion as Transfer and Generation from an Interlinguia with Quality Enhancement through Off-line Human Interaction*, Machine translation Summit, Singapore, 1999.

[6] *EnConverter Specification*, version 3.3, UNL Center/UNDL Foundation, Tokyo 150-8304, Japan 2002.

[7] *UNDL Foundation: The Universal Networking Language (UNL) specifications*, version 3.2., Japan 2003.

[8] *DeConverter Specification*, version 2.7, UNL Center, UNDL Foundation, Tokyo 150-8304, Japan 2002.

[9] Ali, M., and Ali, M. M., *Development of machine translation dictionaries for Bangla Language*, Proceedings of 7th International Conference on Computer and Information Technology (ICCIT), pp. 272-276.

[10] Saha, G. K., *The E2B machine translation: A new approach to HLT. Ubiquity archive*, Association of Computing Machineries (ACM) 6(32), New York, 2005.

[11] Uddin, M. G., Ashraf, H., Kamal. A. H. M, and Ali, M. M., New parameters for Bangla to English statistical machine translation. Proceedings of 3rd International Conference on Electrical & Computer Engineering, 2004, pp. 545-548.

[12] Shahidullah, D. M., *Bangala Vyakaran*, Maola Brothers Prokashoni, Dhaka, August 2003, pp.110-130.

[13] Shniti Kumar, D. C., *Vasha-Prokash Bangla Vyakaran*, Rupa and Company Prokashoni, Calcutta, July 1999, pp.170-175.

[14] Rameswar, D. S., *Shadharan Vasha Biggan and Bangla Vasha*, Pustok Biponi Prokashoni, November 1996, pp.358-377.

[15] Azad, H., *Bakkotottoy*, Second edition, Dhaka-1994.

[16] Karim, M. A., Kaykobad, M., and M. Murshed, *Technical Challenges and Design Issues in Bangla Language Processing*, IGI Global, Disseminator of Knowledge, 2013, pp. 35-78.

[17] Bondopoddaye, H., *Bongio Shobdokosh*, Shahitto Okademy, Calcutta-2001.

[18] Asaduzzaman, M. M., and Ali, M. M., *Morphological analysis of Bangla Words for Automatic Machine Translation*, International Conference on Computer and Information Technology, Dhaka, Bangladesh, 2013, pp. 271-276.

[19] Khairunnahar, K., *Morphological Analysis of Bangla Prefix*, The Dhaka University Journal of Linguistic, 1(2), 2008, pp. 157-168.

[20] Ali, M. N. Y., Sorwar, G., and Shamsujjoha, M., *Formation of Word Dictionary of Bangla Vowel Ended Roots for First Person for Universal Networking Language*, the 2015 International Conference on Information and Knowledge Engineering, Las Vegas, USA.

[21] Ali, M. N. Y., Sorwar, G., Toru, A. and Islam, M. A., and Shamsujjoha, M., *Morphological Rules of Bangla Repetitive Words for UNL Based Machine Translation*, 6th International Conference on Swarm Intelligence, ICSI, Beijing, China, June 25-28, 2015, pp-401-408.

[22] Parikh, J., Khot, J., Dave, S., and Bhattacharyya, P., *Predicate Preserving Parsing*, Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, India.

[23] Earley, J., *An Efficient Context Free Parsing Algorithm*, Communications of the ACM, 1970.

[24] Ali, M. N. Y., Ali, M. A., Nurannabi, M. A., and Das, J. K., *Algorithm for Conversion of Bangla Sentence to Universal Networking Language*, International Conference on Asian Language Processing, IALP, , IEEE Computer Society, Harbin, China, 2010, pp. 118-121.

[25] UNDL, available at http://www.undl.org/, Last access: 02-11-2016.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
VOLUME 5, NO. 1
ISSN 2054-7390

Stock Recommendations using Bio-Inspired Computations on Social Media

[1]Sophia Swamiraj and [2]Rajkumar Kannan
Department of Computer Science, Bishop Heber College (Autonomous), Tiruchirappalli, India.
[1]ssophiababu@gmail.com, [2]rajkumar@bhc.edu.in

ABSTRACT

The tremendous growth of the social networks has paved way for social interactions of investing communities about a company's stock performance. Investors are able to share their comments on stocks using social media platforms. These interactions are captured and mined to produce advice on investing which helps retail investors to do prospective investments to increase profits. In this paper, we propose a novel stock recommendation methodology using Ant Colony Optimization (ACO). This method extracts sentiments from the investor's stock reviews and performs the sentiment analysis, which is optimized by the ACO. This method helps to find the correlation between sentiments and stock values, to make future stock predictions and to give stock recommendations to the retail investor.

Keywords: Stock micro blogging, stock investment, recommendations, user generated content, opinion mining, swarm intelligence.

1 Introduction

Investors require as much information as possible about what's going on in the stock market. They need to do a lot of investigation before making an investment. They have to make the right choice of their investment depending upon their situation and requirements. A lot of small investors do it alone. They do their own research. Novice investors don't know where to begin or more particularly how to screen for stocks. In order to select an individual stock as an investment, investors first need a good source of potential investments. Investors need advice or trading recommendations on which stocks to buy and sell which accomplishes all the above.

The Internet, as a whole, has turned out to be an enabler that aggregates crucial information for stock investor decision making. It is altering how information is passed on to investors and the ways in which investors can take action upon that information [1]. In essence, it changes the way that investors invest trade, obtain and share information [2]. Initially, it was more of combining public information such as public news, financial data, and market updates. More recently, with the arrival of WEB 2.0 and social media [3], user generated content (UGC) are integrating private information in addition to public information [4]. Consequently we study how such virtual investing communities (VIC) as Yahoo Finance and Raging Bull are issuing relevant and valuable UGC data such as investment advice and proprietary analysis. UGC in these channels enriches investors' capability in making better investing decisions by

DOI: 10.14738/tmlai.51.2537
Publication Date: 16th January, 2017
URL: http://dx.doi.org/10.14738/tmlai.51.2537

letting investors to observe the thought process and decision makings of others. Thus it is essential for researchers and practitioners to comprehend how individuals in virtual communities interact with one another and how these behaviours relate to future predictive outcomes.

Social networking sites such as Facebook and Twitter collectively have hundreds of millions of users around the world, and are therefore an excellent venue for investors to disseminate their stock tips.

Recently social investing communities are slowly emerging as a platform to unite investors, help them to share information about stocks, and get them to work collectively to make more informed investing decisions. Several such investing decisions are aggregated to form a collective decision. Investment decisions are shared in the form of opinions. This research proposes an approach that takes investors' collective intelligence through their interactions with the contents, their contributions and finally suggests best investment recommendations.

Internet today is becoming more and more interactive and networked. Web 2.0 platforms such as social networks, wikis, and weblogs empower the people and transform their way of organizing within communities. Without any central control, self organizing groups arise in which members work together by following easy rules of communication and interaction. This new form of cooperation emerging from collective intelligent behaviour changes the process of information sharing and opinion formation [5]. In contrast to the top-down approach of information dissemination by classic media [6], opinions are transferred and formed in the bottom-up approach of Web 2.0 by social swarming.

Understanding the process of opinion formation in human swarms provides great potential for opinion research. Opinion development in swarms can be predicted or might even be manipulated. A new approach is introduced which by using text mining techniques identifies the opinions of single swarm members and analyzes opinion formation with regard to the underlying swarming behaviour by applying methods associated with swarm intelligence. Swarm intelligence being a discipline of artificial intelligence and intends at developing algorithms based on the swarming behaviour of social insects [7]. A new algorithm inspired by the commonly known ant colony optimization meta-heuristic is presented which allows the prediction of opinion development in human swarms. The approach is demonstrated by an exemplary online community in which opinions on stock market are exchanged and discussed.

2 Related work

2.1 UGC in Marketing

Our study closely associates an active area of research from the marketing discipline relating consumer behaviour to economic outcome. It is popular for scholars in this area to study consumer behaviour in the forms of customer ratings or electronic Word-of-mouth (eWOM), user reviews and blogs are very popular areas of study for scholars. The Internet's capacity to reach out to vast audience at low cost has presented a new significance for Word-of-mouth (WOM) as a means to influence and build trust [8].

2.2 UGC in Virtual Investing Communities

Virtual investing communities (VIC) are a reputable social media for online investors. It has blossomed with the growth of the Internet and its reputation stems from offering an environment where investors can collaborate and discuss, monitor what others are doing, or simply to seek out fellowship [9]. We peruse a few samples of studies undertaken to understand the relationship between behaviour of community participants and stock market outcomes.

One of the earlier studies in this area is from [10], which used a sample of 3,000 stocks on Yahoo! message board, and found that earlier day returns, changes in trading volume, and changes in previous day postings have no predictive capacity on stock returns. He found that an increase in volume of overnight postings correlated to a 0.18% average abnormal return. In adding up, he concluded that total posting volume is higher for firms with high short-seller action, accounting performance, excessive past stock returns and, higher price earnings and book-to-market ratios, higher past volatility and trading volume, higher market analyst following, and lower institutional holding [10]. In another study [4], using 181,000 postings from RagingBull.com found that, in general, message board activity does not forecast industry-adjusted returns or abnormal trading volume. However they found that it is likely to predict the number of postings using earlier day's trading volume, number of postings and weighted opinion [4].

A well-referenced paper [11], uses 1.5 million postings from Yahoo! Finance and RagingBull.com message boards, found important but negative simultaneous correlation between number of postings and stock returns on the next day. The return, however, is reasonably very small in comparison to transaction costs. Nevertheless, message posting actions do help to predict volatility and trading volume. In addition, the authors concluded that volume of postings is positively linked with volatility and bullishness. Similarly in [12], apart from verifying that day traders are noise traders, also found that day-trading volume increases volatility but concluded no predictive relationship with stock returns. Das and Chen [13] developed a methodology using five classifier algorithms to mine sentiment from stock message boards but found no significant predictive relationship between sentiment and stock prices. However, consistent with result of [11], [13] reaffirmed the reality of a noteworthy correlation between posting volume and volatility but asserted that sentiment does not predict stock movements. Interestingly, Das et al. [14], found that sentiment does not predict returns but instead returns drive sentiments. They inferred that members of virtual community are more likely to extrapolate past returns rather than to be contrarian, which ultimately leads to a behaviour consistent with the representativeness heuristic [14], [15], [16].

Sabherwal et al. [17] downloaded 160,000 postings from TheLion.com stock message board and conducted an event study to assess daily abnormal returns. The authors found that posting volume positively correlates with stock's abnormal returns on the same day and also forecast next day's abnormal returns. They concluded that online investors focused on sparsely traded micro-cap stocks with low institutional assets and low analyst coverage.

Overall, although significant relationship exists amid VIC activities such as posting volume and stock market movements, previous literature has yet to establish subsistence of any predictive power between sentiment and stock market outcome. This is the research gap we get to answer through investigating the relationship between sentiments of stock micro blog postings with prospective stock price movements.

2.3 UGC in Micro blogging

Although micro blogging is an emerging UGC channel, a few scholars have attempted to examine the relationship among predictors mined from micro blogs with future outcomes such as movie revenue, events and stock prices. For instance, Bollen et al. [18] extracted six mood dimensions from over 9 million Twitter postings using an extended version of Profile of Mood States (POMS). They combined mood components on a on a daily scale and evaluated them to the timeline of cultural, social, economic and political events in the same time period. They found significant relationship between extracted mood dimensions and those happening events. Bollen et al. [19] further widened their prior study in Bollen et

al. [18] specifically towards forecasting DJIA index over the same time period and concluded an accuracy of 87.6%. Another instance is Asur and Huberman [20] which extracted sentiment from 6 million Twitter postings to predict box office income for movies. They benchmarked versus Hollywood Stock Exchange (HSX) and achieved an accuracy of 0.94.

Research correlating predictors in micro blogging with future outcomes is still in its early years. In this study we seek to understand the relationship between sentiments of stock micro blogs with future stock price performances.

2.4 Agent based models

Agent-based models offer computational models to simulate how communications among individuals lead to the emergence of a group organization [21]. It is based on the findings that the behaviour of a group cannot be explained by the independent behaviour of individuals ([22], [23]). Interaction between the individuals results in a high-level organization which crystallizes without awareness of the individuals. Often, they are even unable to assess their own actions and opinions ([24], [25]). Agent-based models are applied to explain the cooperative behaviour in the fields of organization, contagion and cooperation [21]. For instance, Schelling [26] simulates how individual movements according to neighbourhood similarity lead to the development of segregated groups. Axelrod [27] shows how the individuals' adoption of neighbourhood behaviour brings forth global polarization. Berger and Heath [28] study how ideas are spread during discussion boards depending on environmental signals. Rosenkopf and Abrahamson [29] express how innovations diffuse across groups with regard to reputational and informational influences. Sakamoto et al. [30] examine how individual choices develop under varying group influences within online communities. The method presented in this paper can be considered as an agent-based model which simulates opinion formation by social swarming. The method of opinion formation is also based on the interactions among individuals and the orientation towards neighbouring discussion partner. Alternatively, our approach differs in aim and method. We aim to address the phenomenon of opinion formation by using an ant algorithm from swarm intelligence.

2.5 Swarm intelligence

Swarm intelligence offers problem-solving algorithms which are inspired by the swarming behaviour of animals and which can be transcribed to human behaviour. Swarm intelligence consists of two major meta-algorithms: Ant colony optimization and Particle swarm optimization [7]. Ant colony optimization is motivated by the foraging behaviour of ants. It enables the incremental explaining of discrete optimization problems. Digital ants find solutions by following pheromone trails which indicate the quality of an uncertain solution. Particle swarm optimization imitates the behaviour of birds searching for food. It allows population-based solving of continuous optimization problems. Solutions correspond to birds (particles) which are flying around the solution space by following the best birds so far. Two types of algorithms have been developed in the past for solving conventional optimization problems such as time scheduling and route planning. These days new challenges such as data mining and Web mining are being faced. Lots of papers describe ant colony optimization algorithms and particle swarm optimization algorithms for data clustering [31], data classification [32], feature selection [33] and fuzzy-rule induction [34]. Web mining involves the development and application of such algorithms in a few researches. Abraham and Ramos [35] present a cluster algorithm for identifying Web usage patterns. Ujin and Bentley [36] propose an algorithm which leads online shoppers and visitors to interesting Web sites by personal

recommendations based on their preferences. Jensen [33] describes an algorithm for categorizing Web pages based on their topic. Palotai et al. [37] introduce an algorithm which finds news on the Internet. On the other hand, so far there are no algorithms for analyzing the swarming behaviour of Web users during the evolutionary process of opinion formation.

3 Proposed methodology

The proposed approach aims at optimizing various opinions from the individual investors using ACO for stock recommendation where the accurate predictions and recommendations are accomplished by extracting sentiments from the opinions by classifying the individual opinions and optimizing them. The architecture of the proposed approach is shown in figure 1.

The proposed approach consists of two important steps:

1. Opinion classification
2. Opinion optimization using ACO

The algorithmic steps of the proposed approach are as follows.

1) Collecting the tweets
2) Pre-processing the text - tokenize into words, removing special chars, stopwords, stemming, bigrams, trigrams, n-grams
3) Sentiment classification of the pre-processed tweets based on word features.
4) The classified tweets are given as inputs to Ant algorithm.
5) The ant algorithm optimizes the classified tweets to give "buy", "sell" and "hold" signals.

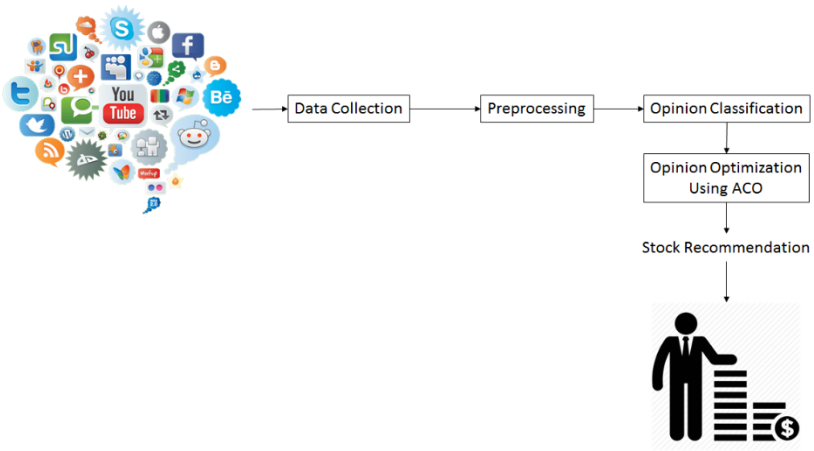

Figure 1. Architecture diagram of the proposed approach

3.1 Opinion classification

The goal of opinion mining is to identify the attitude of single swarm members towards an entity mentioned in postings. Attitudes are classified according to their polarity as “positive”, “negative”, or “no opinion”.

There are two steps involved in opinion mining, the extraction of features from the text and the application of a learning algorithm to identify the polarity of the text [38]. The extraction of features comprises a collective linguistic and statistical analysis. First a posting is decomposed into single words. After removing insignificant words (e.g. “the”), the remaining words are reduced to their stem and their

frequency is calculated. Those word stems which are especially typical for each of the classes (meaning that they appear often in one class but not often in others) are used as the features of the postings. Based on the extracted features, the postings are classified as positive or negative by a learning algorithm. In general, machine learning provides three categories of learning algorithms [39]. Supervised learning algorithms use input and required output data for learning to produce the accurate output data. Reinforcement learning algorithms are trained to generate actions by getting rewards and punishments [40]. Unsupervised learning algorithms, in contrast, accept no feedback and find patterns within the input data which can be used to produce output data [39]. Supervised and unsupervised learning methods are often used for text classification [41]. Supervised learning requires more effort for pre-classifying texts (desired output) but enables enhanced classification results. It is, hence, employed in this approach.

Different supervised learning methods such as Naïve Bayes or Maximum Entropy can be used for text classification [42]. Support Vector Machines [43] are applied because of their ability to process a large number of features and their success in related projects [42]. Their input are sample data records which consist of various discussion postings with their features and classes. Support Vector Machines learn the constraints of a rule by analyzing the sample data which categorizes the postings best. The rule allows a binary classification. If there are three classes, three rules must be learned: “positive” versus “not positive”, “negative’” versus “not negative”, and “opinion” versus “no opinion”. A posting will be assigned to the class which has the highest probability. In the simple two-dimensional case the rule can be described as a straight line (linear rule). Postings lying on one side of the line belong to the first class and those lying on the other side belong to the second class.

3.2 Opinion analysis using ant colony optimization approach

Complex tasks such as picking up objects or finding food can be achieved by colonies of social insects like ants, bees, wasps, and termites by means of cooperation. Swarm intelligence is the collective intelligent behaviour emerging from relatively simple interactions among colony members [44].

In general terms, swarm intelligence can be defined as an occurrence which arises from the social structure of interacting agents over a period of time if the sum of the problems solved collectively is higher than the amount of the problems solved individually [45]. Two preconditions must be satisfied in order for swarm intelligence to develop: The agents must interact with each other and must be capable of problem-solving [45]. Characteristics of emerging swarm intelligence are self-organization, robustness, and flexibility [44]. The members of the swarm interact without supervision or centralized control. The swarm is capable of achieving its task even if some members fall short and is able to adapt to a changing environment.

This phenomenon of collective intelligence is observed not only in the colonies of social insects but also in collaborative groups of humans. By, inspiring one another, correcting mistakes, and exchanging experiences collaborative groups are in a better position of solving problems than individuals [46]. Collaboration can be understood as an act of collective information processing [47]. Discussion is one of its basic forms [45]. Due to the collective process of exchanging information and opinions during a discussion, the total of the combined knowledge of the community becomes more valuable than the sum of the knowledge of all individual community members [48]. Web 2.0 platforms increase this effect of knowledge enhancement [49]. A wider range of people can connect more easily and more swiftly to reach a common opinion in an online discussion.

The Web also offers a benefit for opinion research. The process of opinion formation can be traced by applying mining techniques. A novel approach based on text mining and swarm intelligence is presented which is capable of analyzing the evolutionary method of opinion formation by social swarming. Text mining enables the recognition of opinions of single community members. An algorithm connected with swarm intelligence, especially the commonly known ant colony meta-heuristic, permits the prediction of the opinion trend during the collective intelligent process of opinion formation in online communities.

The aim of opinion analysis is to gain a better understanding of opinion formation in social swarms. With this knowledge opinion trends can be predicted and the process of opinion formation might even be manipulated.

Opinion analysis is inspired by the collective intelligent behaviour of living ants finding the shortest path between their nest and their source of food. This intelligent behaviour emerges from the ants' indirect manner of communicating by leaving and following pheromone trails in their environment – a phenomenon called stigmery [50]. While ants are moving about they drop chemical substances called pheromones on their paths. The more the same path is frequented, the more the pheromone intensity increases and the more likely this path will be followed by other ants. If the same path is only followed by a few ants, the pheromone intensity of the path decreases due to evaporation. As a result of this feedback loop, the probability that the path will be followed by an ant depends on the number of ants having taken this path before.

The ants' behaviour seems to resemble in some ways the behaviour of human swarming within online communities and can be used as a simplified model to simulate the process of opinion formation. Members of online communities communicate indirectly with each other by posting messages to a discussion thread. In their postings they can express positive or negative opinions. The more messages of the same opinion are posted, the more other people are attracted by this opinion and the more likely they are to follow this opinion.

The collective intelligent behaviour of ant colonies is also the basic idea of the ant colony optimization meta-heuristic, from which an algorithm for simulating the process of opinion formation can be derived. In ant colony optimization algorithms, possible solutions for a given problem are represented by paths [51]. If a path is followed by an ant, a certain amount of pheromones is deposited on it depending on the quality of the solution. Evaporation gradually decreases the pheromone amounts on those paths which are not traversed frequently. This means that the corresponding solution is not particularly appreciated.

When simulating the process of opinion formation the problem is to predict the polarity of the next posted opinion in a discussion thread. Possible solutions are represented by two different paths: one for positive and one for negative opinions. An ant predicts the next posted opinion by following the corresponding path and drops a certain amount of pheromones on this path depending on the correctness of the prediction. Evaporation depends on the sequence of postings in the thread and leads to a reduction of the pheromone amount on the path of the less frequently mentioned opinion. The ant is more likely to predict the opinion class whose corresponding path has a higher amount of pheromones.

In general, the ant colony optimization meta-heuristic comprises the following components [44]:

- A heuristic function which evaluates the quality of the solution found by an ant
- A rule for pheromone updating which describes how to reinforce pheromones on paths

- A rule for pheromone evaporation which specifies how pheromones on paths diminish over time
- A decision function which finds solutions by considering the value of the heuristic function and the amount of pheromones on paths.

In order to build up an algorithm based on ant colony optimization for predicting opinions in online swarms, these components must be specified and integrated into a procedure.

Figure 2 shows the flowchart of the developed procedure. First, all variables are initialized. The pheromone values of both paths representing the positive and negative opinions are given equal amounts of pheromones. While the discussion is going on, the opinion trend in terms of the next posted opinion class is predicted by an ant. The ant predicts the opinion class by choosing a path according to the decision function. As soon as the next message is posted to the thread its content is checked. If no opinion is expressed in the posting evaporation takes place. However, if the posting contains an opinion the correctness of the ant's prediction is evaluated. In case the predicted opinion differs from the posted opinion, the average error ratio is increased. Otherwise the average error ratio is decreased. Thereafter, the heuristic values of the decision function are adjusted depending on the dynamics of the opinion discussion. In addition, the pheromone value of the path representing the predicted opinion is updated. Finally, evaporation takes place which decreases the pheromone values of all paths.

According to the procedure, the functions for decision making, pheromone updating and evaporation have to be defined. The decision function determines the predicted opinion at time *i* by comparing the weighted sum of pheromones on the positive (τ_p^i) and negative path (τ_n^i). It is implemented as a signum function based on the difference of the weighted pheromone sums of both paths (positive and negative opinions), By this means the opinion of the path with the highest weighted sum of pheromones is predicted.

$$di = sgn\left(n_i^p \tau_i^p\right) - \left(n_i^n \tau_i^n\right) \tag{1}$$

If the decision function yields the value 1, a positive opinion is predicted by the ant. In the case of value -1 a negative opinion is forecasted. The pheromone sums of both paths are weighted by the iteratively computed heuristic values η_p^i and η_n^i. The heuristic values are incremented by a certain factor x if the actual opinion (O_i) equals the previous one ($O_i - 1$) and decremented by x if both opinions are different. By doing so, the dynamics of opinion changes during opinion prediction is taken into account.

$$\eta_i = \begin{cases} \eta_i - 1 + \ \ x, if\ O_i = O_i - 1 \\ \eta_i - 1 - \ \ x, if\ O_i \neq O_i - 1 \end{cases} \tag{2}$$

If a lot of consecutive messages with no opinion are posted to the thread, the influence of the last posted opinions on the future opinion trend becomes insignificant. In such a case, the prediction should be based on all opinions posted to the thread. In the ant algorithm evaporation leads to a rapid diminution in pheromone values of the negative and positive path if there is a sequence of postings without opinions. A minimum-rule derived from the Max-Min Ant System of Stützle and Hoos [52] is implemented to change the prediction basis in this case. According to this rule, the decision function predicts the future opinion trend based on the opinion class most frequently mentioned in the thread, if the pheromone values of both paths fall below a minimum value. The rule is also inspired by the biological archetype of behaviour

of real ants. If the amount of pheromones on a path cannot be smelled any more, the ants rely on their instinct when choosing a path.

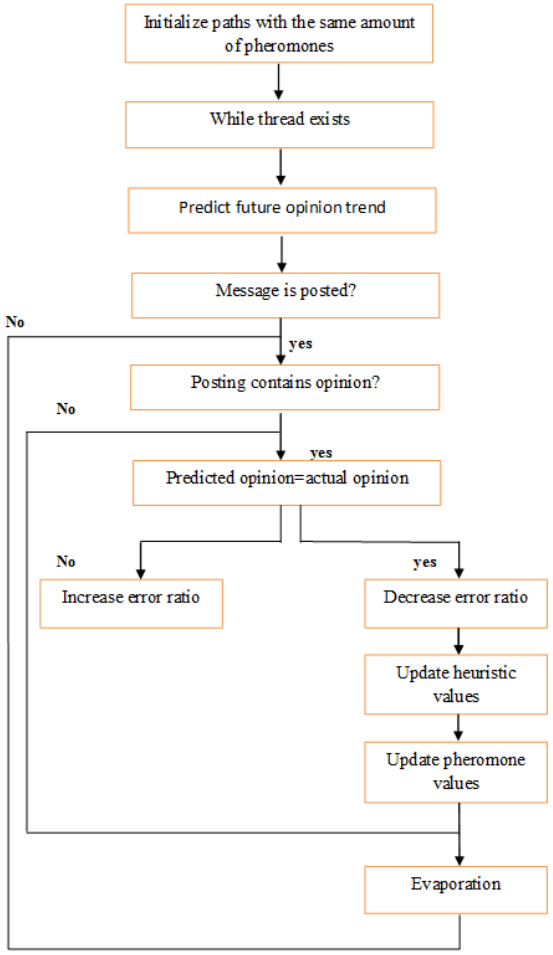

Figure 2. ACO Algorithm Flowchart

In order to reinforce predicted opinions the pheromone values of the corresponding paths are updated. Pheromone updating is realized by adding a predefined amount of pheromones ρ to the current pheromone value of the selected path.

$$\tau_i = \tau_i + \rho \tag{3}$$

Since the opinion trend in a discussion can change from time to time more weight should be added to recent opinions than to past opinions. Pheromone evaporation enables this weighting. It is realized by multiplying the pheromone value τ_i with a certain factor e<1.

$$\tau_{i+1} = \tau_i e \tag{4}$$

The above equations are followed from [53] which incorporate an algorithm based on ant colony optimization for predicting opinions in online swarms.

4 Experiments and Results

4.1 Data

The primary data for this study was downloaded from Stocktwits.com (http://www.stocktwits.com) for the period May 21 2016, to April 30, 2016 (22 days). We obtained over 7,140 stock micro blog postings for the company NETFLIX.

Stock micro blog postings were pre-processed; those without any ticker, more than one ticker, or not in NASDAQ exchange were removed leaving 2,140 valid postings for testing and 5000 postings for training. A list of top 6 stock tickers with corresponding number of postings is shown in table 1 while a description of all attributes is in table 2. Interestingly, top 10% of all the stock tickers are responsible for over 70% of all postings. These are popular stocks, consistent with the finding that people invest in the familiar while often ignore principles of portfolio theory [54].

Table 1. Distribution of postings by top 6 tickers.

S. No	Ticker	Total	Exchange
1	AAPL	7212	NASDAQ
2	AMZN	6220	NASDAQ
3	BBRY	2502	NASDAQ
4	GOOG	1803	NASDAQ
5	MSFT	1576	NASDAQ
6	NFLX	7140	NASDAQ

Table 2. Description of posting attributes [55]

Variable	Description
Sentiment	0-neutral , 1-bullish, -1-bearish (manually labelled)
Posting	Posting id, post date, day of the week, time of the day, market hours, text of posting.
Author	Expert, bio, url, location, follower, following, total postings, posting per day, retweet, direct, mention, etc.
Ticker	Exchange, volume, past 7 days closing prices and volumes.
Market	Past 7 days NASDAQ index.

4.2 Experiments on opinion classification

For validation, opinions posted to the stock investment community of Stocktwits.com were classified. The result of opinion classification is shown in the table 3. Stocktwits.com is the online platform of stock investment community. 5000 postings were extracted and assigned to the three classes “positive”, “negative”, and “no opinion” by a human annotator.

In order to examine the classification results a stratified ten-fold cross validation is applied. This means that all postings are divided into ten equally sized parts containing the same proportions of class labels. There are ten validation loops. In each loop nine parts are used for learning the classification rules and the remaining part for testing the classification rules learned. After ten runs the average precision and recall are calculated. While precision describes how many of the recognized opinions are correct, recall shows how many of the opinions are really recognized. The results of validation are shown in table 4. They indicate that learning was more successful for positive opinions and negative opinions than for neutral

opinions. Lessons learned from misclassification show that quite often postings are not recognized as neutral if they contain several positive arguments or negative information but a neutral introduction or a neutral conclusion. This problem is planned to be solved by attaching more weight to the words at the end and the beginning of the postings.

Table 3. Results of opinion classification

Class	No of tweets	Accuracy
Positive	1379	100%
Negative	170	86%
Neutral	591	67%

Table 4. Precision and recall

Class	Precision	Recall
Positive	0.586	0.898
Negative	0.724	0.104
Neural	0.576	0.368

4.3 Experiments on opinion analysis and recommendations

In order to validate the ant algorithm for opinion prediction the financial communications platform for the investing community stocktwits.com was analyzed. Three companies APPLE, NETFLIX and GOOGLE in which community members discussed their opinions on stocks were extracted. All opinions mentioned in the postings were classified as 'positive', 'negative' or 'no opinion'. Before applying the ant algorithm the parameters of the functions involved must be determined. Tests with different combinations of parameters revealed the following best set:

- Pheromone update ρ: 0.5
- Heuristic factor x: 0.3
- Minimum threshold: 0.0001
- Evaporation rate e: 0.8

Based on this set of parameters, the average error ratio is measured over the entire period of time. It is calculated as the fraction of all incorrectly predicted opinions to all opinions. Table 5 depicts average ratios for the three company discussions. The low error rate of 17.2% for the company "AAPL" indicates high prediction accuracy. The error rate for the company "GOOG" is higher indicating that prediction was less successful.

Table 5. Average error ratios

Discussion on Company	Amount of postings	Average error ratio
$APPL	1242	17.2%
$NFLX	2140	13.26%
$GOOG	2460	41.3%

Besides the average error ratio of the entire period, the development of the error ratio over time is important as well. It reveals whether the decision function enables incremental learning. Figure 3 and 4 depict the error curves associated with the discussion on the company "NFLX". The descending error curve

indicates a successful learning process. This effect results from the heuristic values which adapt incrementally to the dynamics of the discussion.

The necessity of the minimum-rule becomes apparent when looking at the development of the pheromone amounts on the positive and negative paths over time. For example, Figure 5, 6, 7 and 8 show the pheromone curves associated with the discussion on the company "NFLX". 12.5% of the pheromone values fell below the minimum threshold so that prediction was based on the most frequent opinion class.

In addition, the pheromone development shows the underlying swarming behaviour of opinion formation. The amounts of pheromones decrease over time which indicates a levelling in discussion. At the same time opinions are getting more homogeneous. During discussion a clear opinion trend emerges from the initially differing opinions of the swarm members. At the end of the discussion there are only a few opinions differing from the overall opinion trend which have less influence. This can be interpreted as a sign of robustness of the swarming behaviour. Application results of the ant algorithm show that prediction is more successful for discussion threads in which the length of the sequences of equal opinions vary to a high degree.

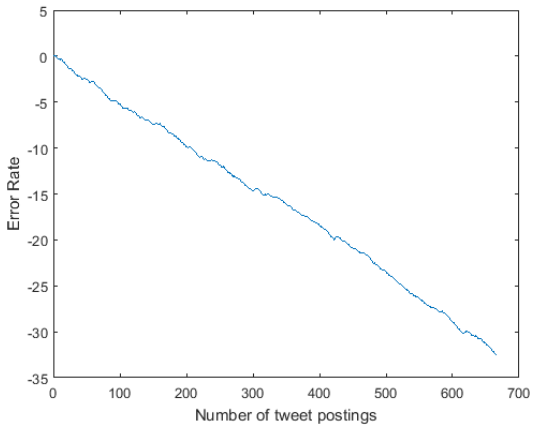

Figure 3. Normal Ant error ratio graph- 666 tweets - $NFLX

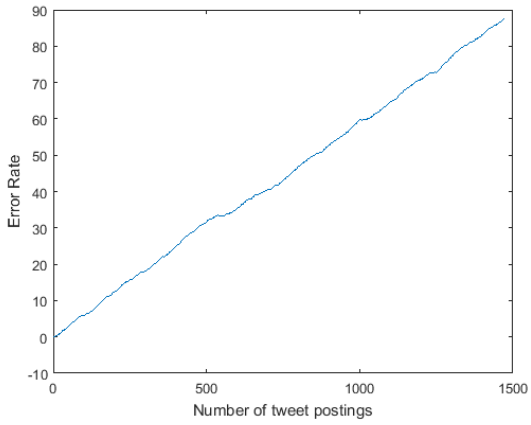

Figure 4. Normal Ant error ratio graph-1475 tweets - $NFLX

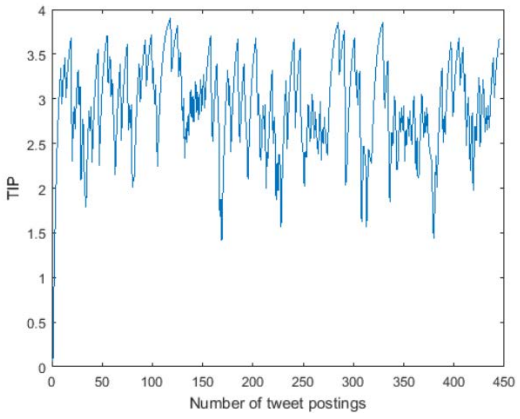

Figure 5. Pheromone graph TIP - 666 tweets

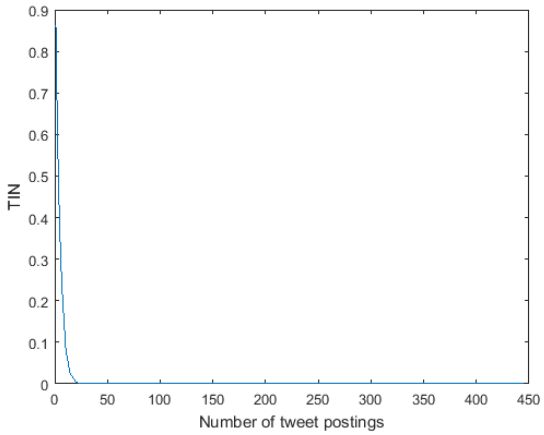

Figure 6. Pheromone graph TIN – 666 tweets

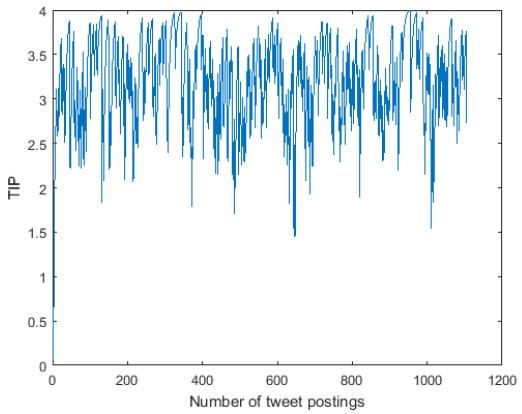

Figure 7. Pheromone graph TIP -1475tweets

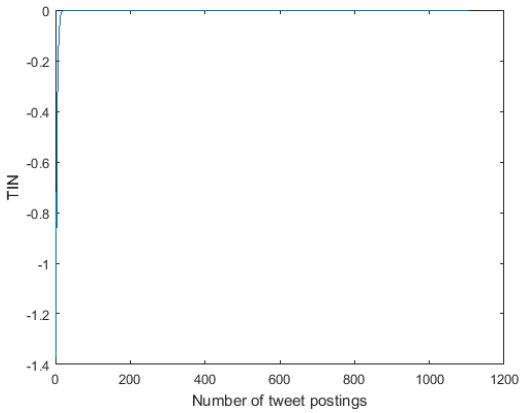

Figure 8. Pheromone graph TIN-1475 tweets

Table 6. Ant colony algorithm results for stock tweets of Netflix Inc.

		Ant algorithm output		
Time	**Total tweets**	**Positive**	**Negative**	**Neutral**
April 23,24,25	666	444	1	221
April 26,27,28,29	1474	0	1105	370

Table 7. Netflix Historical Stock Prices

Date	Open	High	Low	Close/last	Volume	*Adj close
Apr 29, 2016	90.50	90.56	88.21	90.03	13,968,000	90.03
Apr 28, 2016	91.50	92.67	90.09	90.28	11,474,900	90.28
Apr 27, 2016	92.18	92.50	90.21	91.04	12,218,900	91.04
Apr 26, 2016	93.50	93.55	91.25	92.43	15,330,900	92.43
Apr 25, 2016	95.70	95.75	92.80	93.56	14,985,400	93.56
Apr 22, 2016	94.85	96.69	94.21	95.90	15,806,300	95.90
Apr 21, 2016	97.31	97.38	94.78	94.98	19,919,400	94.98

As we observe from the above two tables 6 and 7 the ant's prediction proved to be accurate. The historical stock prices for the company Netflix are increasing for the dates April 22-25 for which in the above table 6 the ant result gives a positive sentiment prediction. For the dates April 26, 27, 28, and 29 the historical stock prices are declining and the ant accurately gives a negative sentiment prediction. Thus we are able to get the accurate sentiment predictions to give the accurate stock recommendation to "sell" the Netflix company stocks during this period of decline. Hence we prove the sentiments collected from the stock related tweets are able to guide the retail investor's decision making into profitable investments. Stock recommendations thus guided by sentiments are very useful for retail investors and to everyone in the investing community to increase profits and to decrease losses.

5 Conclusion

The outcomes are more accurate recommendations which best suit an individual trader amongst the multiple choices. Recommender systems can maximize investment returns in stock portfolio investments. Investment returns are enhanced with a reduction in trading losses using intelligent recommender systems. This research will result into a recommender system that allows the retail investor to save a lot of time in locating potentially profitable trading opportunities. The capability to scan the universe of stocks and only select the ones that meet their criteria in a matter of seconds is a huge advantage for the active

trader. Stock investing recommender system provides the professional help to small investors so that they can select the right stocks for them and get good returns. Investors get ahead in investing; they not only earn great returns in the bull market but also minimize losses in the bear market.

REFERENCES

[1] Barber, B. M., & Odean, T. (2001). The internet and the investor. *The Journal of Economic Perspectives*, *15*(1), 41-54.

[2] Zhang, Y., & Swanson, P. E. (2010). Are day traders bias free?—evidence from internet stock message boards. *Journal of Economics and Finance*, *34*(1), 96-112.

[3] Ullrich, C., Borau, K., Luo, H., Tan, X., Shen, L., & Shen, R. (2008, April). Why web 2.0 is good for learning and for research: principles and prototypes. In *Proceedings of the 17th international conference on World Wide Web* (pp. 705-714).

[4] Tumarkin, R., & Whitelaw, R. F. (2001). News or noise? Internet postings and stock prices. *Financial Analysts Journal*, *57*(3), 41-51.

[5] Rheingold, H. (2007). Smart mobs: The next social revolution. Basic books.

[6] Kolbitsch, J., & Maurer, H. A. (2006). The Transformation of the Web: How Emerging Communities Shape the Information we Consume. *J. UCS*, *12*(2), 187-213.

[7] Blum, C., & Li, X. (2008). Swarm intelligence in optimization. In *Swarm Intelligence* (pp. 43-85).

[8] Dellarocas, C. (2003) "The Digitization of Word of Mouth: Promise and Challenges of Online Feedback Mechanisms" *Management Science,* 49(10):1407-1424.

[9] Wasko, M. & Faraj, S. (2005) "Why Should I Share? Examining Social Capital and Knowledge Contribution in Electronic Networks of Practice" *MIS Quarterly,* 29(1), 35-57.

[10] Wysocki, P. D. (1998). Cheap talk on the web: The determinants of postings on stock message boards. *University of Michigan Business School Working Paper*, (98025).

[11] Antweiler, W., & Frank, M. Z. (2004). Is all that talk just noise? The information content of internet stock message boards. *The Journal of Finance*, *59*(3), 1259-1294.

[12] Koski, J. L., Rice, E. M., & Tarhouni, A. (2004). Noise trading and volatility: Evidence from day trading and message boards. *Available at SSRN 533943*.

[13] Das, S. R., & Chen, M. Y. (2007). Yahoo! for Amazon: Sentiment extraction from small talk on the web. *Management Science*, *53*(9), 1375-1388.

[14] Das, S., Martinez-Jerez, A. & Tufano, P. (2005) "eInformation: A Clinical Study of Investor Discussion and Sentiment" *Financial Management,* 34(3), 103-137.

[15] Lakonishok, J., Shleifer, A., & Vishny, R. W. (1994). Contrarian investment, extrapolation, and risk. *The journal of finance*, *49*(5), 1541-1578.

[16] Kahneman, D., & Tversky, A. (1973). On the psychology of prediction. *Psychological review*, *80*(4), 237.

[17] Sabherwal, S., Sarkar, S. K., & Zhang, Y. (2008). Online talk: does it matter?. *Managerial Finance*, *34*(6), 423-436.

[18] Bollen, J., Pepe, A. & Mao, H. (2010a) "Modeling public mood and emotion: Twitter sentiment and socioeconomic phenomena" *Proceedings from 19th International World Wide Web Conference* Raleigh, North Carolina.

[19] Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, *2*(1), 1-8.

[20] Asur, S., & Huberman, B. A. (2010, August). Predicting the future with social media. *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010,* (Vol. 1, pp. 492-499).

[21] Goldstone, R. L., & Janssen, M. A. (2005). Computational models of collective behavior. *Trends in cognitive sciences*, *9*(9), 424-430.

[22] Axelrod, R. M. (1997). The complexity of cooperation: Agent-based models of competition and collaboration. Princeton University Press.

[23] Salganik, M. J., Dodds, P. S., & Watts, D. J. (2006). Experimental study of inequality and unpredictability in an artificial cultural market. S*cience*, *311*(5762), 854-856.

[24] Loewenstein, G., & Schkade, D. (1999). Wouldn't it be nice? Predicting future feelings. *Well-being: The foundations of hedonic psychology*, 85-105.

[25] Bonds-Raacke, J. M., Fryer, L. S., Nicks, S. D., & Durr, R. T. (2001). Hindsight bias demonstrated in the prediction of a sporting event. *The Journal of social psychology*, *141*(3), 349-352.

[26] Schelling, T. C. (1971). Dynamic models of segregation†. *Journal of mathematical sociology*, *1*(2), 143-186.

[27] Axelrod, R. (1997). The dissemination of culture a model with local convergence and global polarization. *Journal of conflict resolution*, *41*(2), 203-226.

[28] Berger, J. A., & Heath, C. (2005). Idea habitats: How the prevalence of environmental cues influences the success of ideas. *Cognitive Science*, *29*(2), 195-221.

[29] Rosenkopf, L., & Abrahamson, E. (1999). Modeling reputational and informational influences in threshold models of bandwagon innovation diffusion. *Computational & Mathematical Organization Theory*, *5*(4), 361-384.

[30] Sakamoto, Y., Sadlon, E., & Nickerson, J. V. (2008). Bellwethers and the emergence of trends in online communities. In *Proceedings of the 30th Annual Conference of the Cognitive Science Society* (pp. 1416-1421).

[31] Ye, F., & Chen, C. Y. (2005). Alternative KPSO-clustering algorithm. In *淡江理工學刊*, *8*(2), 165-174.

[32] Admane, L., Benatchba, K., Koudil, M., Siad, L., & Maziz, S. (2006). AntPart: an algorithm for the unsupervised classification problem using ants. *Applied Mathematics and Computation*, *180*(1), 16-28.

[33] Jensen, R. (2006). Performing feature selection with ACO. In *Swarm Intelligence in Data Mining* (pp. 45-73).

[34] Galea, M., & Shen, Q. (2006). Simultaneous ant colony optimization algorithms for learning linguistic fuzzy rules. In *Swarm intelligence in data mining* (pp. 75-99).

[35] Abraham, A., & Ramos, V. (2003, December). Web usage mining using artificial ant colony clustering and linear genetic programming. In *The IEEE Congress on Evolutionary Computation, 2003. CEC'03.*, (Vol. 2, pp. 1384-1391).

[36] Ujjin, S., & Bentley, P. J. (2003, April). Particle swarm optimization recommender system. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium, 2003. SIS'03.* (pp. 124-131).

[37] Palotai, Z., Mandusitz, S., & Lórincz, A. (2006). Computer study of the evolution of 'news foragers' on the Internet. In *Swarm Intelligence in Data Mining* (pp. 203-219).

[38] Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems* (No. 1). Oxford university press.

[39] Szuba, T. M. (2001). *Computational collective intelligence*. John Wiley & Sons, Inc..

[40] Hofstätter, P. R. (1986). Gruppendynamik, Kritik der Massenpsychologie; 3. *Auflage, Rowohlt, Reinbeck bei Hamburg*.

[41] Smith, J. B. (1994). *Collective intelligence in computer-based collaboration*. CRC Press.

[42] Kolbitsch, J., & Maurer, H. A. (2006). The Transformation of the Web: How Emerging Communities Shape the Information we Consume. *J. UCS*, *12*(2), 187-213.

[43] Johnson, N., Rasmussen, S., Joslyn, C., Rocha, L., Smith, S., & Kantor, M. (1998). Symbiotic Intelligence: self-organizing knowledge on distributed networks driven by human interaction. In *Proceedings of the 6th International Conference on Artificial Life* (pp. 403-407).

[44] Weiss, S. M., Indurkhya, N., Zhang, T., & Damerau, F. (2010). *Text mining: predictive methods for analyzing unstructured information*.

[45] Ghahramani, Z. (2004). Unsupervised learning, Bousquet O., Raetsch G., and von Luxburg U.(Eds.), Advanced Lectures on Machine Learning, LNAI3176.

[46] Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, *4*, 237-285.

[47] Lee, C. H., Yang, H. C., Chen, T. C., & Ma, S. M. (2006, August). A comparative study on supervised and unsupervised learning approaches for multilingual text categorization. In *First International Conference on Innovative Computing, Information and Control-Volume I (ICICIC'06)* (Vol. 2, pp. 511-514).

[48] Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10* (pp. 79-86).

[49] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273-297.

[50] Martens, D., De Backer, M., Haesen, R., Baesens, B., & Holvoet, T. (2006). Ants constructing rule-based classifiers. In *Swarm Intelligence in Data Mining* (pp. 21-43).

[51] Parpinelli, R. S., Lopes, H. S., & Freitas, A. A. (2002). Data mining with an ant colony optimization algorithm. *IEEE transactions on evolutionary computation*, *6*(4), 321-332.

[52] Stützle, T., & Hoos, H. H. (1996). Improving the Ant System: A detailed report on the MAX–MIN Ant System. *FG Intellektik, FB Informatik, TU Darmstadt, Germany, Tech. Rep. AIDA–96–12*.

[53] Kaiser, C., Krockel, J., & Bodendorf, F. (2010, January). Swarm intelligence for analyzing opinions in online communities. In *43rd Hawaii International Conference on System Sciences (HICSS), 2010,* (pp. 1-9).

[54] Huberman, G. (2001). Familiarity breeds investment. *Review of financial Studies*, *14*(3), 659-680.

[55] Oh, C., & Sheng, O. (2011). Investigating predictive power of stock micro blog sentiment in forecasting future stock price directional movement, In *Proceedings of ICIS 2011*. 17.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
VOLUME 5, NO. 1
ISSN 2054-7390

Kannada Named Entity Recognition and Classification using Support Vector Machine

[1]S Amarappa, [2]S V Sathyanarayana
[1]Department of TCE, Jawaharlal Nehru National College of Engineering, Shimoga - 577 204, India;
[2]Department of E & C, Jawaharlal Nehru National College of Engineering, Shimoga - 577 204, India;
amarappas@yahoo.com; svs.tce@gmail.com

ABSTRACT

Named Entity Recognition and Classification (NERC) is a process of identification of proper nouns in the text and classification of those nouns into certain predefined categories like person name, location, organization, date, time etc. Kannada NERC is an essential and challenging work which aims at developing a novel model based on Support Vector Machine. In this paper, tf-idf and POS features are used, which are extracted from a training corpus created manually. Furthermore, the model is trained and tested with different kernels: polynomial, rbf, sigmoid and linear kernels. The details of implementation and performance evaluation are discussed. The experiments are conducted on a training corpus of size 1, 51,440 tokens and test corpus of 7,000, 11,000, 15,000, 20,000, 30,000, 40,000 and 50,000 tokens. It is observed that the model works with an average precision, recall and F1-measure of 87%, 88% and 87.5% respectively for a linear kernel SVM on the test corpus of 7,000 tokens.

Keywords: Natural Language Processing; Hyperplane; Support vectors; Named Entity Recognition; Classification; Support vector machine; Training Corpus; Test Corpus.

1 Introduction

Natural Language Processing (NLP) has two major tasks: Natural Language Understanding (NLU) and Natural Language Generation (NLG) [1]. NLU deals with machine reading comprehension, i.e., the level of understanding of a text or message. NLG is the task of generating natural language from a machine representation system such as a knowledge base. Apart from NLG and NLU, the other tasks to be done in NLP include automatic summarization, Information Extraction (IE), Information Retrieval (IR), Named Entity Recognition (NER) etc.

In NLP, the primary goal of IE and IR is to automatically extract structured information. NERC is a typical subtask of IE [2]. NERC involves processing of structured and unstructured documents and identifying proper names that refer to persons, organizations, locations (cities, countries, rivers, etc.), date, time, etc. The aim of NERC is to automatically extract proper names that are useful to address many problems such as Machine Translation, Information Extraction, Information Retrieval, Question Answering, and Automatic Text Summarization etc., [3].

India has more than 1,652 mother tongues of which 22 are scheduled languages included in the Constitution. Among the 22 scheduled languages, Kannada is one of the major Dravidian languages of

DOI: 10.14738/tmlai.51.2549
Publication Date: 12th February, 2017
URL: http://dx.doi.org/10.14738/tmlai.51.2549

India, spoken predominantly in the state of Karnataka. The Karnataka official language Act 1963 recognized Kannada as its official language. Kannada, whose native speakers are called Kannadigas (Kannadigaru) number roughly 40 million, making it the 33rd most spoken languages in the world ("Census 2001: Languages by state". censusindia.gov.in. Retrieved on 12 February 2013).

1.1 Kannada language features

The language uses forty-nine phonetic letters, divided into three groups: swaragalu (vowels – thirteen letters); vyanjanagalu (consonants – thirty four letters); and yogavaahakagalu (neither vowel nor consonant - two letters: the anusvara and the visarga), similar to the vowels and consonants of English. The character set is almost identical to that of other Indian languages. This language is inflected with three genders (masculine, feminine, and neutral) and two numbers (singular and plural). The Noun is inflected by various factors such as case, number and gender. It is a free-word order language with rich heritage and large grammar.

1.2 Challenges and Issues specific to Kannada language

Kannada is one of the many Indian languages, presenting a large set of complications. Processing of Kannada language and extraction of named entities on the phrasal semantics basis is challenging because of the reasons:

- Kannada is a highly agglutinating and inflected language.
- Kannada language has no capitalization.
- It has a Brahmi script with high phonetic characteristics that could be utilized by NERC system.
- There is non-availability of large gazetteer, lack of annotated data, lack of standardization and spelling.
- There are a number of frequently used words (common nouns), which can also be used as names.
- These nouns act as adjectives in many contexts and handling these nouns carefully is very much essential. Phrasal semantic analysis of these nouns is interesting.
- As there is lack of annotated data, the whole corpus is annotated by hand. While annotating, care is taken on overlaps among types of Named Entities (NEs). NE overlaps of this kind are carefully tagged based on the phrasal context.

Examples of NE overlaps are:

- Common noun vs. proper noun: 'surya' which means sun may be person's name.
- Organization vs. person name: 'TaTa', person name as well as an organization name.
- Organization vs. place name: Mumbai meets Chennai at Bangalore. Here 'Mumbai' and 'Chennai' are names of playing teams rather than the names of cities.
- Person name vs. place name: The word 'kashi' is used as a person name as well as the name of a place.

1.3 Motivation

From the survey carried out in Section 2, it is observed that a lot of work on NERC has been done in English and other foreign languages. NERC work in Indian languages is still in its initial stage. As far as Indian languages are concerned, some works related to NERC are found in Hindi, Bengali, Telugu, Tamil, Oriya, Manipuri, Punjabi, Marathi and Assamese languages. But in Kannada language, NERC work is not yet reported except our former works using Hybrid approach [27], Hidden Markov Model (HMM) [28],

Multinomial Nave Bayes (MNB) classifier [29] and Continuous Random Fields (CRF) [30]. The experimental results of all the methods are encouraging; nevertheless the works on NERC in Kannada are to be investigated and implemented with different statistical approaches apart from HMM, MNB and CRF. This has motivated us to take up NERC in Kannada using Support Vector Machine (SVM) classifier as the projected research paper.

1.4 Novelty in this work

With the challenges and issues in Kannada language, we propose the application of Support Vector Machine to resolve the problem of the NERC for Kannada language. We find the work carried out has novelty factors in many respects as mentioned here under:

- The main contribution is that there has been no SVM method available for Kannada language; therefore we ought to deal with the problem from the scratch. This is the first solution of its kind to the problem of NERC using SVM for Kannada language.
- Support Vector Machine model was already used to solve the NERC problem in other languages, but we have to deal with the effort of creating an annotated dataset for the previously neglected language.
- As the annotated Kannada corpus (Unicode) is not available, the whole raw corpus that we have shaped is manually tagged and is checked by local linguistic experts. While annotating we have used fine-grained tags as mentioned in IJCNLP-2008 NERSSEAL shared task data set.
- The work is explained in detail and furthermore, it provides an interesting view over the status of the art with respect to NLP solutions for Indian languages.
- The essence of this work is, tuning up of the SVM idea to the Kannada language NERC.
- The language text is not transliterated (unlike the NERC in other Indian languages) to Roman; instead it is honestly taken from Unicode text files typed by us and trained our model. The test data is also taken from Unicode text files.
- We have used the document classification perception for individual tokens, treating them as independent documents. The features extracted from the training corpus include ‘tf-idf’ features and parts of speech tags. From these features Support Vector Machine hyperplane is estimated.

This contribution towards Kannada NLP is expected to be a motivation for young researchers and for the readers interested in Information Extraction from natural languages. The application of the results of this work is relevant mostly to research, dealing with Indian languages. So, the work is definitely relevant as it boosts up the scientific developments related to the processing of the Kannada language. Furthermore, the proposed solution was experimentally tested with a variety of test-set sequences and the results are encouraging.

The paper presents in detail the implementation and evaluation of a solution for Named Entity Recognition based on Support Vector Machine for the Kannada language. The results obtained from the proposed model are quite encouraging with an average accuracy of 87% for a linear kernel. The rest of this paper is organized as follows: Section 2 discusses about the details of existing work. Section 3 deals with Support Vector Machine principles which are used for NERC in the paper. The proposed methodology and implementation details are dealt in Section 4. The SVM classifier’s evaluation measurements are discussed in Section 5. Finally the results are evaluated and discussed in Section 6 followed by conclusions in Section 7.

2 Existing work

The NLP started way back in the 1940s and from then to 1980s, the NLP systems were based on complex sets of hand-made rules. After 1980s, machine learning algorithms were used in NLP research and recent NLP algorithms are based on statistical machine learning. The term Named Entity was introduced in the sixth Message Understanding Conference (MUC-6) [4]. The different techniques for addressing the NERC problem include: Hidden Markov Models (HMM) (D.Bikeletal, 1997), Decision Trees (S.Sekine, 1998), Maximum-Entropy Models (ME) (A.Borthwick, 1998), Support Vector Machines (SVM) (M.Asahara & Matsumoto, 2003), and Conditional Random Fields (CRF) (A.McCallum & Li, 2003) [31].

A lot of NLP work has been done in English, as there is an enormous amount of data available in it. A voluminous work is done in most of the other European languages, some of the Asian languages like Chinese, Japanese, Korean and other foreign languages like Arabic, etc. NLP research in Indian languages is at the initial stage, as annotated corpus and other lexical resources have started appearing recently. In computational linguistics, Kannada is lagging far behind, compared to other Indian languages. In the following paragraphs, we present a brief survey of research on NERC in Indian languages including Kannada. This is not a comprehensive and thorough survey, but is an indication of current status in NERC research.

Few works on NER in English language are: In [12] the authors have built a CRF based NER system that achieves 91.02% F1-measure on the CoNLL 2003 dataset. An overview of the techniques employed to develop domain specific NER systems is dealt in [13]. In [14] the authors have devised an unsupervised NER by generating Gazetteers and resolving ambiguity.

In [5] the authors have developed an algorithm for rule based NER in Urdu. In [6] the authors carried out a work on Person Name Entity Recognition for Arabic. Reference [7] discusses about SVM based language independent NER. Reference [8] discusses about the first step towards Assamese NER. In [9] the authors have developed a system using CRF approach for NER in Bengali and Hindi. In [10] the authors have developed NER system for Bengali. Reference [11] discussed about Bengali NER using SVM.

In [15] the authors have developed a system for NER in Hindi using Max-Entropy and Transliteration. In [16] the authors have developed Hindi NER by aggregating rule based heuristics and HMM. Reference [17] discusses a composite kernel for NER. In [18] the authors have experimented NER using HMM on Hindi, Urdu and Marathi languages. Reference [19] gives introduction to the CoNLL- 2003 shared task a language-independent NER. Reference [20] discussed about a language independent NER system for Bengali & Hindi using SVM. Reference [21] deals with a model on CRF based NER in Manipuri. Reference [22] deals with SVM based NER for Manipuri. Reference [23] presents the construction of a hybrid, three stages NER for Tamil. In [24] the authors have developed a tourism domain focused NER for Tamil using CRF. Reference [25] describes a Max-Ent, NER system for Telugu. Reference [26] discusses about Telugu NER using language dependent features and rule based approach.

In Kannada language, the only papers available are [27], [28], [29] and [30]. In [27] the authors have carried out Named Entity Recognition Classification and Extraction (NERCE) for Kannada language using hybrid approach, which combines man made rules and Hidden Markov Model (HMM) on a small training corpus of 10,000 tokens and text corpus of 1000 tokens and the experimental results are good with an average F1-measure of 94.85%. In [28] the authors have carried out NERC using Hidden Markov Model

(HMM) on a small training corpus of 10,000 tokens and text corpus of 1000 tokens and the experimental results are encouraging with an average F1-measure of 86%. In [29] we have carried out Kannada NERC based on Multinomial Naïve Bayes (MNB) Classifier and achieved an average F1-measure of 81% on a training corpus of 95,170 tokens and test corpus of 5,000 tokens. In [30] we have carried out Kannada NERC based on Conditional Random Fields (CRF) and achieved an average F1-measure of 82% on a training corpus of 95,127 tokens and test corpus of 5,000 tokens.

3 Support Vector Machine

Although the details of Support Vector Machine (SVM) are well established in the literature, we reiterate the information essential to our research. Support Vector Machine (SVM) is a supervised learning method used for binary classification, regression and outlier's detection. SVM has a simple structure and is derived from statistical learning theory by Vladimir Vapnik and his colleagues in 1992. Given some data points, each belonging to one of two classes and the goal is to decide to which class a new data point belongs. In SVM, a data point is viewed as an n-dimensional vector in n-dimensional space $\mathbf{V = R^n}$ and we want to know whether we can separate such points with an (n - 1) dimensional hyper plane (canonical plane). There are many hyperplanes that might classify the data but, the best one is with the largest margin. A hyperplane is a subspace of one dimension less than its ambient space. A hyperplane of an n-dimensional space V is a subset with dimension n-1 in V that separates the space into two half spaces. The hyper plane is found by using a subset of training points in the decision function called support vectors, and the margin. To find the margin, two parallel supporting planes are constructed, one on each side of the canonical plane.

3.1 Multi-class SVM

Multiclass classification aims at classifying data points belonging to more than two classes with the assumption that each sample is assigned to one and only one label.

The dominant approach for multiclass classification is to reduce the single multiclass problem into multiple binary classification problems. Common approaches of reducing a multiclass problem into multiple binary classifiers include:

- One-versus-the-rest also known as one-versus-all strategy aims at fitting one classifier per class. If there are n-classes of data points then for each classifier, the class is fitted against all the other n-1 classes and hence it requires n classifier models to be trained. Since each class is represented by one and one classifier only, it is possible to gain knowledge about the class by inspecting its corresponding classifier. This is the most commonly used strategy and is a fair default choice.
- One-versus-one approach (Knerr et al., 1990) constructs one classifier per pair of classes. At prediction time, the class which received the most votes is selected. If n is the number of data classes, then n * (n - 1) / 2 classifiers are to be constructed and each one trains data from two classes. Since it requires to fit n * (n - 1) / 2 classifiers, this method is usually slower than one-vs-the-rest approach. The basic principle of a binary SVM classifier is derived from the geometrical equation of a straight line y = mx+b, thus defining a linear discriminant function,

$$g(x) = w^T X + b \tag{1}$$

In the Equation (1), w = [w_1, w_2]; w_1 and w_2 are weights of x and y respectively, with 'b' being the intercept to y-axis. The weights w_1 and w_2 may be positive or negative. X is a vector in two-dimensional space. The

line $w^T X + b = 0$, is used as a hyperplane in two class classification problems. In SVM, the hyperplane should be found such that it maximizes the margin separating the positive training points from the negative training data points. The Lagrange multiplier of Equation (2) is used to obtain optimized hyperplane, where the term $½||w||^2$ should be minimized, subject to the constraints $Y_i (w^T X + b) \leq 1$.

$$L_P(w, b, \alpha_i) = \frac{1}{2} ||w||^2 - \sum_{i=1}^{n} \alpha_i (Y_i (w^T + b) - 1) \quad (2)$$

Such that $\alpha_i \geq 0$. Solving the Equation (2), gives w_1, w_2, b, and α_i. These parameters determine a unique maximal margin solution. The two parallel positive class and negative class supporting planes are constructed, one on each side of the hyperplane by the constraints:

$$w^T X + b = +1 \quad (3)$$

$$w^T X + b = -1 \quad (4)$$

4 Proposed work and Methodology

The main aim of this work is to develop a supervised statistical machine learning NERC system for Kannada language based on SVM. NERC involves identification of proper names in texts, and classification of those names into a set of pre-defined categories of interest such as: person names (names of people), organization names (companies, government organizations, committees, etc.), location names (cities, countries etc.), and miscellaneous names (date, time, number, percentage, monetary expressions, number expressions and measurement expressions). The functional block diagram of the proposed system is as shown in Figure 1.

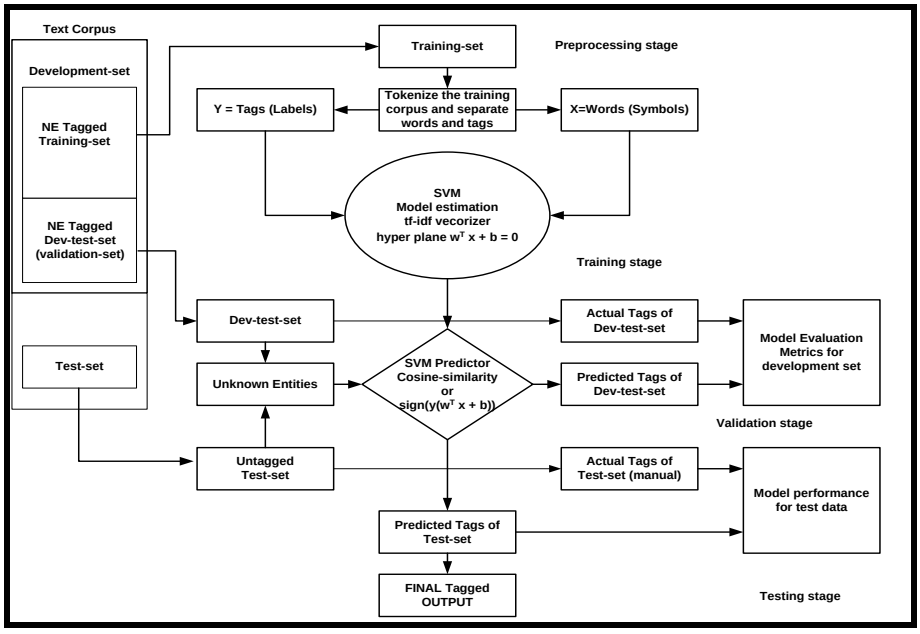

Figure 1: SVM model for Kannada NERC

This Section deals with the design and development of NERC system based on the SVM model. We resent the details of the methodology, design, and development of the proposed system. The machine learning used in the work is fully supervised SVM. The features extracted from the training corpus include POS tag

features, term frequency features and inverse document frequency features. The model is trained and tested with different kernels: polynomial, rbf, sigmoid and linear kernels.

4.1 Corpus creation and usage

Kannada NERC is very hard without tagged corpus and hence we manually tagged about 150K Kannada words. This Kannada corpus is used to build the NERC Model. The manually tagged corpus includes: Part of EMILLE (Enabling Minority Language Engineering) corpus [http://www.ciil.org/Schemes.aspx (Linguistic Data Consortium for Indian Languages (LDCIL))], a part of the corpus taken from web articles and part of the corpus extracted from Kannada books. The entire corpus is tagged based on the phrasal semantics taking the text context into consideration. The whole corpus is divided into two sets: Development-set and Test-set as shown in Figure 1. First, select the Development-set and then subdivide it into the Training-set and development test set (Dev-test- set). The Training-set is used to train the model and the Dev-test-set is used to perform error analysis. The Test-set serves for the final evaluation of the system. The machine learning used in the work is fully supervised SVM.

The data set is annotated by four annotators with a common agreement. Tag set is chosen exactly similar to that of tag set used in IJCNLP-2008 NERSSEAL shared task data set. While annotating the corpus manually, Named Entity (NE) overlaps are carefully tagged based on the phrasal context. Examples of NE overlaps are mentioned in Section 1.2. The corpus created is having 70% to 80% of distinctive words of Kannada language.

For a binary SVM classifier the input training-set consists of 'N' number of data points in the form $\mathbf{(X_i, y_i)}$ where, $X_i \in (X_1, X_2 \ldots X_N)$ and $y_i \in (+1, -1)$. Moreover X_i is a point in two dimensional vector space $\mathbf{(X_i \in R^2)}$ and represents the contextual information of the tagged word.

In this paper we have used thirteen Named Entities (NEs) with twenty two tags as indicated in Table 1. A non-named entity is tagged as 'NONE' with label '22'. Person name has 4 tags, where the tag NEP is used for person names having only one word in it. If person name is of two words, first word is tagged with NEPB and second word with NEPE and further if person name is of more than two words, first word is tagged with NEPB, last word is tagged with NEPE and intermediate words are tagged with NEPI. The same rules are followed for locations and organizations also (Table 1).

The sample of training corpus is: Training-set X = [(amar, NEP), (jnnce, NEO), (shimoga, NEL), (shimoga, NEL), (jnnce, NEO), (Davanagere, NEL), (shimoga, NEL), (pesitm, NEO), (sathyanarayana, NEP)].

4.2 Pre-processing stage

The tagged training text corpus is tokenized into words (symblols) and tags (states / classes). The separated words are X = [w1, w2, w3 … wN] and separated tags are Y = [y_1, y_2, y_3 … y_N]. Each tag in Y is assigned a number called label, i.e., $y_i \in (0, 1, 2 \ldots 22)$ (or $y_i \in (c_0, c_1 \ldots c_{22})$) as given by:

['NEP:0', 'NEPB:1', 'NEPI:2', 'NEPE:3', 'NEL:4', 'NELB:5', 'NELI:6', 'NELE:7', 'NEO:8', 'NEOB:9', 'NEOI:10', 'NEOE:11', 'NED:12', 'NETE:13', 'NETP:14', 'NETO:15', 'NEB:16', 'NEM:17', 'NEN:18', 'NETI:19', 'NEA:20', 'NE:21', 'NONE:22'].

For the sample training-set mentioned in Step 1, tokenize and separate words and tags/labels: Separated words: X = [amar, jnnce, shimoga, shimoga, jnnce, davanagere, shimoga, pesitm, sathyanarayana] and Separated tags/labels: Y = [NEP:0, NEO:8, NEL:4, NEL:4, NEO:8, NEL:4, NEL:4, NEO:8, NEL:4].

Table 1. Named Entity Tag set

Named Entity (NE)	Tag	Tag Lalel	NE Meaning	Example	Example meaning
Person	NEP	0	Name of a person having only one word	ಈಶ್ವರಚಂದ್ರ/NEP	Eshwarachandra
	NEP BIE	1,2,3	Name of a person Begin Intermediate End	ಮೋಹನ್/NEPB ದಾಸ್/NEPI ಗಾಂಧಿ/NEPE	Mohan Das Gandhi
Location	NEL	4	Name of a place/location having only one word	ಶಿವಮೊಗ್ಗ/NEL, ಕರ್ನಾಟಕ /NEL	Shivamogga, Karnataka
	NEL BIE	5,6,7	Name of a person Begin Intermediate End	ಯುನೈಟೆಡ್/NELB ಸ್ಟೇಟ್ಸ್ಆಫ್ /NELI ಅಮೇರಿಕ/NELE	United States of America
Organization	NEO	8	Name of an organization having only one word	ನಗರಸಭೆ/ORG	Municipality
	NEO BIE	9,10,11	Name of a organization Begin Intermediate End	ಭಾರತೀಯ /NEOB ವಿಜ್ಞಾನ/NEOI ಸಂಸ್ಥೆ /NEOE	Indian Institute of Science
Designation	NED	12	Designation	ಜೆನರಲ್ಮ್ಯಾನೇಜರ್, ಕಮಿಷ್ನರ್/NED	General Manager, Commissioner
Term	NETE	13	Terms, Diseases	ಸಿದ್ಧಾಂತ, ನಿಯಮ, ಕಾಲರ/NETE	Theory, Rule, Cholera
Title Person	NETP	14	Title Person	ಡಾ॥, ಶ್ರೀ, ಶ್ರೀಯುತ	Dr||, Mr, Mr
Title Object	NETO	15	Title Object	ಕುರ್ಚಿ, ಮೇಜು	Chair, Table
Brand	NEB	16	Brand Name	ಪೆಪ್ಸಿ, ಕೋಲ	Pepsi, Cola
Measurement	NEM	17	Measurement	೪,೫೦೦ರೂ, ೫ಕೆ.ಜಿ.	4,500 Rs, 5 Kg
Number	NEN	18	Number	೩.೧೪, ೪,೫೦೦	3.14, 4,500
Time	NETI	19	Date, Time etc.,	೩ನೇ ಸೆಪ್ಟೆಂಬರ್ ೧೯೯೧	3rd Septembar 1991
Abbreviation	NEA	20	Abbreviation	ಎನ್ಎಲ್ ಪಿ, ಬಿಜೆಪಿ	NLP, BJP
Noun entity	NE	21	Common nouns	ಕತೆಗಾರ	Writer
Not a NE	NONE	22	Not a NE	ಮಳೆ, ಹೋಗು, ಹೈ	rain, go, hi

We are using the concept of document classification where, each input word is treated as a document and tag as its class: for the above example we have the details as mentioned in Table 2.

Table 2. Each word is a document

Document No.	word (x)	tag (y)
D1	amar	NEP

D2	jnnce	NEO
D3	shimoga	NEL
D4	shimoga	NEL
D5	jnnce	NEO
D6	davanagere	NEL
D7	shimoga	NEL
D8	pesitm	NEO
D9	sathyanrayana	NEP

4.3 Training stage

The various steps of training the model are as given below:

3.1 Input to the training stage are X and Y

3.2 The model finds important words (vocabulary features) by removing repeated words and stop words from X. The model also finds unique tags/ labels from Y. The vocabulary feature words are W = [w_1, w_2, w_3 … wn] and unique tags are Y = [y_1, y_2, y_3 … y_k].

For the sample training-set mentioned in Step 1, the important words (vocabulary features) by removing repeated words and stop words are: W = [w_1: amar, w_2: jnnce, w_3: shimoga, w_4: davanagere, w_5: pesitm, w_6: sathyanarayana] and the unique tags/labels are Y = [NEP: 0, NEO: 8, NEL: 4].

3.3 Find raw count of each vocabulary word of W in Training-set, i.e., term frequencies tf.

3.4 The term-frequency is a measure of how many times a particular term of W, is present in the document of Training-set X=[x_1, x_2 … x_N] (or D= [d_1, d_2 … d_N]). In our model, each word of X is treated as a document (word x_1 = document D_1). The term-frequency is defined as a counting function and is given in Equation (5).

$$\mathrm{tf}(t, d) = \sum_{x \in d} \mathrm{fr}(x, t) \tag{5}$$

Where fr(x, t) is a simple function, defined by Equation (6).

$$\mathrm{fr}(x, t) = \begin{cases} 1, & \text{if } x = t \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

The $\mathrm{tf}(t, d)$ returns count of t in document d. The $\mathrm{tf}(t, d)$ in matrix form is denoted by Equation (7).

$$M_{|D| \times F} = (M_{\mathrm{train}}) \tag{7}$$

3.5 Find inverse document frequency $\mathrm{idf}(t)$ of training corpus defined by the function $P(t|d) = \frac{|\{d:t \in d\}|}{|D|}$, so idf is define as Equation (8).

$$\begin{aligned} \mathrm{idf} &= -\log P(t|d) \\ &= \log \frac{1}{P(t|d)} \\ \mathrm{idf}(t) &= \ln\left(\frac{|D|+1}{1+|\{d:t \in d\}|}\right) + 1 \end{aligned} \tag{8}$$

Here $|\{d: t \in d\}|$ is the number of documents where the term t appears; when the term-frequency function satisfies $\mathrm{tf}(t, d) \neq 0$. It should be noted that adding 1 into the formula above avoids zero division.

3.6 Now to find tf-idf use the following steps tf-idf is found using Equation (9).

$$3.6.1.\ \mathrm{tf-idf} = \mathrm{tf}(t, d) \times \mathrm{idf}(t) \quad (9)$$

3.6.2. Find idf for each feature present in the feature matrix with the term frequency and idf weights can be represented by a vector as given by Equation (10).

$$\overrightarrow{idf}_{train} = [idf(t_1),\ idf(t_2) \ldots idf(t_k)] \quad (10)$$

3.6.3. tf-idf matrix of training set in un-normalized form is found by:

Now the tf matrix, $\mathrm{M}_{|\mathrm{D}|\times\mathrm{F}} = \mathrm{M}_{\mathrm{train}}$ of Equation (7) and the idf matrix $\overrightarrow{\mathrm{idf}}_{\mathrm{train}}$ of Equation (10) are multiplied to calculate the tf-idf weights.

3.6.4. And then multiply $\mathrm{M}_{\mathrm{idf}}$ to the term frequency matrix, so the final result can be defined as Equation (11).

$$[M_{ft-idf}]_{ixk} = [M_{train}]_{ixk} \times [M_{idf}]_{kxk} \quad (11)$$

3.6.5. tf-idf matrix of Training-set in normalized form is given in Equation (12).

$$M_{ft-idf} = \frac{M_{ft-idf}}{\|M_{ft-idf}\|_2} \quad (12)$$

tf–idf vectors are the actual trained parameters/features of the SVM model (Scikitlearn version 0.14 documentation). The tf–idf vectors and POS tags are the main features that are used to determine the hyperplane weight vectors w_1, w_2 and the intercept '*b*'.

As already mentioned in Table 1, we have used 13 named entities with 22 classes and a non-named entity is assigned a tag 'NONE', and hence 23 classifiers are trained: [SVM_{m0}, SVM_{m1} … SVM_{m21}, SVM_{m22}].

The SVM_{m0} is trained such that it assigns positive value (≥ +1) for class c_0 and negative value (≤ -1) for remaining classes. In general SVM_{mi} is trained to give positive result for class c_i and negative result for rest of the classes.

To predict the class of an unknown (test) feature vector 'p', the classifier uses the separating hyperplane $w_i^T p + b_i = 0$. If $w_i^T p + b_i \geq +1$, then feature vector 'p' belongs to class c_i. Else if $w_i^T p + b_i \leq -1$, then feature vector 'p' does not belongs to class c_i.

4.4 Validation stage

A fold of the tagged training corpus is reserved as Dev-test-set and multiple evaluations are performed on various Dev-test-sets. The scores thus obtained from those evaluations are combined to get the average score. A fold of the annotated training data is taken from the Development-set as Dev-test-set and the following computations are performed:

a) Pre-processing and tf-idf computations are done for Dev-test-set as explained in Steps 2 and 3.

b) The tf-idf vector of each sample of Dev-test-set is given to classifier SVM_{m0}. The classifier SVM_{m0} assigns a positive value (≥ +1) if the sample belongs to class c_0. If the sample doesn't belong to class c_0 it assigns a negative value (≤ -1), and then sample is fed to SVM_{m1}. The classifier SVM_{m1} says whether the sample belongs to class c1 or not. If not the sample is fed to next classifier, and this process is continued till the sample is classified for a right class.

Figure 2 shows the tree of SVM decoding. Here values ≥ +1 are normalized to +1 and values ≤ -1 are normalized to -1.

c) From the actual class labels and predicted class labels of Dev-test set, find precision, recall and F1-measure. Repeat this process on all folds of Dev-test-set and calculate average F1-measure thus validating the model.

4.5 Testing stage

Test-set is taken from the corpus set. The computations are performed similar as in validation stage:

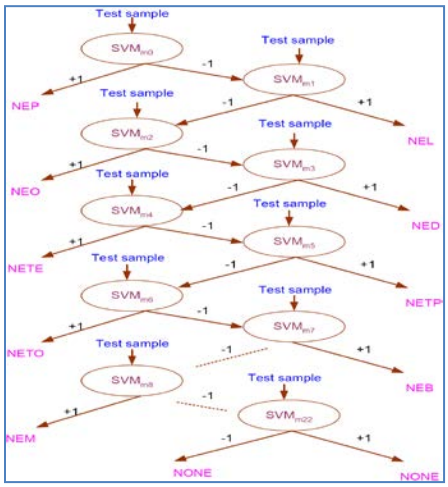

Figure 2. Multi-class SVM for named entity classification

The following algorithm gives the implementation procedure of the SVM Model:

Algorithm:

1. Reading the tagged corpus from the directory and dividing into 10-folds
   corpus ← read tagged corpus
   corpus size ← count of tokens in whole corpus
   folds ← divide the corpus size into ten equal folds
2. 10-fold cross validation of the model
   tag_set = [NEP, NEL, NEO, NED, NETE, NETP, NETO, NEB, NEM, NEN, NETI, NEA, NE, NEPB, NEPI, NEPE, NELB, NELI, NELE, NEOB, NEOI, NEOE, NONE]
   tag_set_labels = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22]
   Begin
   (i) Preprocessing
      development test set ←reserve ith fold of training
      corpus training set ← take the remaining nine folds as training set
      words ← separate words of training set
      tags ← separate tags of training set
      labels ← assign labels to the tags of training set
   (ii) Feature extraction and training of SVM model
      vectorizer ← TfidfVectorizer

time0 ← read system time

training words ← transform words of step (i) into vectors using vectorizer

classifier ← svm.SVC(kernel='linear')

(kernels used are: linear, rbf, sigmoid and poly)

SVM classifier ← input training words and labels to the classifier

time1 ← read system time

training time ← time1 – time0

(iii) Testing with reserved fold of the development test set

development test set (DTS) ← take reserved i^{th} fold of training corpus

words of DTS ← separate words of development test set

actual tags of DTS ← separate tags of development test set

actual labels of DTS ← assign labels to the tags of development test set

time0 ← read system time

test words of DTS ← transform words of DTS into vectors using vectorizer

SVM predicted labels of DTS ← SVM classifier receives test words of DTS as input

time1 ← read system time

fold test time ← time1 – time0

(iv) Evaluation metrics

precision ← precision score from actual labels & SVM predicted labels of DTS

recall ← recall score from actual labels & SVM predicted labels of DTS

f1 ← f1 score from actual labels & SVM predicted labels of DTS

class report ← class report from actual labels & SVM predicted labels of DTS

End

Combine the scores of all the ten folds for the cross validation.

3. Testing of the SVM model

(i) Train the SVM model for all the 10 folds of training corpus as explained in (ii) of Step 2

(ii) Testing with Test-set

test set ← read untagged test corpus from test corpus root directory

words of test set ← words of test set

actual tags of test set ← find actual tags of test set (manually)

actual labels of test set ← assign labels to the tags of test set

time0 ← read system time

test words of test set ← transform words of test set into vectors using vectorizer

SVM predicted labels ← SVM classifier receives test words of test set as input

time1 ← read system time

test time ← time1 – time0

(iii) Evaluation metrics and classification report are found similar to (iv) of Step 2

5 Performance Evaluation Metrics

It is important to know the quality of the SVM machine learning algorithm. Several statistical measurements can be used to estimate the performance of the algorithm. These measurements are collected from a confusion matrix show in Table 3, which contains information about the real and predicted classifications done by the algorithm (https://en.wikipedia.org/wiki/Precision_and_recall).

True positives (TP) - the number of correct predictions that an instance is positive

True negatives (TN) - the number of correct predictions that an instance is negative

False positives (FP) - the number of incorrect predictions that an instance is positive

False negatives (FN) - the number of incorrect predictions that an instance is negative

The aim of the algorithm is to maximize the TP and true negatives TN predictions. The effectiveness of the algorithm is characterized with the recall and precision measurements.

Table 3. Confusion Matrix

		PREDICTED CLASS	
		YES	NO
ACTUAL CLASS	YES	TP	FN
	NO	FP	TN

Recall (R) is the Sensitivity or True Positive Rate (TPR) that measures the ability of the algorithm to find all relevant entities as given by Equation 13. R = Number of correct answers - produced / Total number of possible - correct answers

$$R\ (TPR) = TP/\ (TP + FN) \tag{13}$$

A high recall score tells that most of the relevant entities were retrieved by the algorithm, while a low recall indicates that the most relevant entities were missed by the algorithm.

Precision (P) measures the ability of the algorithm to retrieve only relevant entities, which is computed using Equation 14.

P = Number of correct answers - produced / Total number of answers – produced

$$P = TP/\ (TP + FP) \tag{14}$$

A high precision score indicates that most of the retrieved entities are relevant. A low precision means that the algorithm cannot distinguish relevant entities while retrieving all entities.

F1-Measure is the efficiency measure that combines recall and precision together.

F1 -Measure is the traditional F1-measure or balanced F1-score.

$$F1 - Measure\ (F1) = 2PR/\ (P + R) \tag{15}$$

$$Accuracy = (TP + TN)/\ (TP + FP + TN + FN) \tag{16}$$

Fallout or False Positive Rate (FPR)

$$FPR = FP/ (FP + TN) \quad (17)$$

Specificity or True Negative Rate (TNR)

$$TNR = TN/ (TN + FP) \quad (18)$$

Miss Rate or False Negative Rate (FNR)

$$FNR = FN/ (FN + TP) \quad (19)$$

$$TPR + FNR = 1 \quad (20)$$

$$TNR + FPR = 1 \quad (21)$$

6 Results and Discussions

The proposed system is designed and implemented as discussed in Section 4. The system is tested using several test cases, containing training corpus of size 1, 51,440 tokens. The test corpus is chosen in such a way that it satisfies the entire phrasal context; which is an inherent feature of Kannada language. It is to be noted that the system achieves an average accuracy of 87% on a test corpus of 7000 tokens with linear kernel. The details of the results obtained are as given below. The system's performance is measured in terms of Precision (P), Recall (R) and F1-measure (F1) as discussed in Section 5. The details of the corpus created in this work are given in Section 4. The nature of input test sequence and output tagged sequence are given in Table 4 and Table 5 respectively. The corpus size and program run time are tabulated in Table 6. Table 7 tabulates the results of 10 fold cross validation where validation fold is of size 15,144 tokens. Table 8 indicates the confusion matrix of the experiment. Table 9 indicates the total count of NE's in the training corpus. Table 10 indicates the final classification results of test-set corpus of size 7000 tokens with linear kernel and Table 11 shows the error analysis.

Table 4. Input test sequence

ವಾಷಿಂಗ್ಟನ್ (ಪಿಟಿಐ): ಮುಂಬರುವ ಲೋಕಸಭಾ ಚುನಾವಣೆ ಬಳಿಕ ನರೇಂದ್ರ ಮೋದಿಯೊಂದಿಗೆ. ಅಮೆರಿಕ ರಾಜತಾಂತ್ರಿಕ ಕೆಲಸ ನಿರ್ವಹಿಸಲು ಸಿದ್ದವಿದ್ದು ಇಲ್ಲಿ ವೀಸಾ ಪ್ರಶ್ನೆಯೇ ಇಲ್ಲ ಎಂದು ಅಮೆರಿಕ ಸ್ಪಷ್ಟಪಡಿಸಿದೆ. ಮೋದಿ ವಿಶ್ವದ ಅತಿ ದೊಡ್ಡ ಪ್ರಜಾಪ್ರಭುತ್ವ ರಾಷ್ಟ್ರದ ನಾಯಕನಾದರೆ ಅವರ ಜೋತೆ ನಾವು ಕೆಲಸ ಮಾಡುತ್ತೇವೆ ಎಂದು ಒಬಾಮಾ ಆಡಳಿತದ ಹಿರಿಯ ಅಧಿಕಾರಿಗಳು ತಿಳಿಸಿದ್ದಾರೆ. ಭಾರತೀಯ-ಜನತಾ-ಪಕ್ಷ ಮೋದಿ ಅವರನ್ನು ಪ್ರಧಾನಿ ಅಭ್ಯರ್ಥಿ ಎಂದು ಘೋಷಿಸಿದೆ. ಒಂದು ವೇಳೆ ಆ ಪಕ್ಷ ಅಧಿಕಾರಕ್ಕೆ ಬಂದರೆ ಮೋದಿಗೆ ಗೌರವ ನೀಡಬೇಕು. ಹಾಗಾಗಿ ಯಾವುದೇ ವೀಸಾ ಸಮಸ್ಯೆ ಉದ್ಭವಿಸುವುದಿಲ್ಲ ಎಂದು ಅಧಿಕಾರಿಗಳು ತಿಳಿಸಿದ್ದಾರೆ. .....

Table 5. Output tagged sequence

ವಾಷಿಂಗ್ಟನ್/NEL (ಪಿಟಿಐ):/NEO ಮುಂಬರುವ/NONE ಲೋಕಸಭಾ/NE ಚುನಾವಣೆ/NE ಬಳಿಕ/NONE ನರೇಂದ್ರ/NEPB ಮೋದಿಯೊಂದಿಗೆ./NEPE ಅಮೆರಿಕ/NEL ರಾಜತಾಂತ್ರಿಕ/NONE ಕೆಲಸ/NONE ನಿರ್ವಹಿಸಲು/NONE ಸಿದ್ಧವಿದ್ದು/NONE ಇಲ್ಲಿ/NONE ವೀಸಾ/NETE ಪ್ರಶ್ನೆಯೇ/NONE ಇಲ್ಲ/NONE ಎಂದು/NONE ಅಮೆರಿಕ/NEL ಸ್ಪಷ್ಟಪಡಿಸಿದೆ./NONE ಮೋದಿ/NEP ವಿಶ್ವದ/NE ಅತಿ/NONE ದೊಡ್ಡ/NONE ಪ್ರಜಾಪ್ರಭುತ್ವ/NEtE ರಾಷ್ಟ್ರದ/NE ನಾಯಕನಾದರೆ/NONE ಅವರ/NONE ಜೋತೆ/NONE ನಾವು/NONE ಕೆಲಸ/NETE ಮಾಡುತ್ತೇವೆ/NONE ಎಂದು/NONE ಒಬಾಮಾ/NEP ಆಡಳಿತದ/NONE ಹಿರಿಯ/NETE ಅಧಿಕಾರಿಗಳು/NE ತಿಳಿಸಿದ್ದಾರೆ./NONE ಭಾರತೀಯ/NEOB ಜನತಾ/NEOI ಪಕ್ಷ/NEOE ಮೋದಿ/NEP ಅವರನ್ನು/NONE ಪ್ರಧಾನಿ/NED ಅಭ್ಯರ್ಥಿ/NE ಎಂದು/NONE ಘೋಷಿಸಿದೆ./NONE ಒಂದು/NEN ವೇಳೆ/NETI ಆ/NONE ಪಕ್ಷ/NE ಅಧಿಕಾರಕ್ಕೆ/NONE ಬಂದರೆ/NONE ಮೋದಿಗೆ/NEP ಗೌರವ/NETE ನೀಡಬೇಕು./NONE ಹಾಗಾಗಿ/NONE ಯಾವುದೇ/NONE ವೀಸಾ/NETE ಸಮಸ್ಯೆ/NONE ಉದ್ಭವಿಸುವುದಿಲ್ಲ/NONE ಎಂದು/NONE ಅಧಿಕಾರಿಗಳು/NE ತಿಳಿಸಿದ್ದಾರೆ./NONE .....

As already mentioned, SVM model of Kannada NERC is trained using a tagged corpus of size 1, 51, 440 tokens. A sample input test sequence is given in Table 4. The SVM model generates the tagged output sequence as shown in Table 5. It can be noted that the Table 4 is a subset of actual test corpus of 7,000 tokens and the corresponding output tagged sequence is also the subset of the actual tagged sequence of 7,000 tokens. The performance of the designed SVM model is measured using various evaluation measurements as discussed in Section 5.

Table 6. Corpus size and program Run time

The training set size for the model	1,51,440 words
Total number of samples treated by the classifier	1,51,440 words
Total number of features extracted by the classifier	33273 (symbols)
Feature extraction Time (Training of SVM model)	1244.594 seconds
The test set size for the model	7000 words
Total number of features of test set	5775 (symbols)
Feature extraction Time for test data	2.031 seconds

Table 7. Results of 10 fold cross validation

FOLDS	Precision %	Recall %	F1 %	Support
1	81	81	81	15144
2	84	83	83.5	15144
3	85	83	84	15144
4	88	87	87.5	15144
5	85	84	84.5	15144
6	88	87	87.5	15144
7	79	77	78	15144
8	77	76	76.5	15144
9	83	82	82.5	15144
10	87	85	86	15144
Average/Total	83.7	82.5	83.1	151440

From Table 6 it can be noted that the execution time depends on the size of the input test corpus sequence. Table 7 shows the scores of individual folds and the combined scores of all the ten folds.

Table 8. Confusion matrix of the experiment

		PREDICTED CLASS																
ACTUAL CLASS		NEP	NEPBIE	NEL	NELBIE	NEO	NEOBIE	NED	NETE	NETP	NETO	NEB	NEM	NEN	NETI	NEA	NE	NONE
	NEP	189	4	2	0	0	0	1	0	0	0	0	0	0	0	0	0	128
	NEPBIE	3	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15
	NEL	1	2	99	0	0	0	0	0	0	1	0	0	0	0	0	2	43
	NELBIE	0	1	1	6	0	0	0	0	0	0	0	0	0	0	0	0	4
	NEO	0	1	0	0	13	3	0	0	0	0	0	0	0	0	0	2	9
	NEOBIE	0	0	1	0	1	9	0	1	0	0	0	0	0	0	0	2	6
	NED	0	2	0	0	0	0	27	0	0	0	0	0	0	0	0	2	13
	NETE	1	0	0	0	0	0	0	24	0	0	0	1	0	3	0	3	71
	NETP	2	0	0	0	1	0	0	0	11	0	0	0	0	0	0	1	12
	NETO	0	1	0	0	2	1	0	5	0	34	0	0	0	0	0	2	54
	NEB	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	7
	NEM	1	0	0	0	0	0	0	0	0	0	0	24	3	1	0	0	39
	NEN	0	2	0	0	0	0	0	0	0	0	0	0	55	1	0	1	92
	NETI	0	1	1	0	0	0	0	0	0	0	0	0	3	38	0	0	12
	NEA	0	1	0	0	0	0	0	0	0	0	0	0	0	0	5	0	5
	NE	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	247	147
	NONE	8	4	2	1	1	5	0	1	6	0	0	4	16	1	0	71	5349

From the confusion matrix, the calculation of P, TPR, and FPR are done as follows:

Consider named entity NEP as Positive class and all others entities as negative class:

TP for NEP class = 189 (first element of primary diagonal)

FP for NEP class = 18 (first column sum excluding TP value of 189)

FN for NEP class = 135 (first row sum excluding TP value of 189)

TN = 5950 (diagonal elements sum of confusion matrix excluding TP value of 189)

Precision (P) = TP / (TP + FP) = 189 / (189+18) = 0.91

Recall(R/TPR) = TP / (TP + FN) = 189 / (189+135) = 189/324 = 0.58

F1-Measure = 2PR / (P + R) = 2x 0.91x 0.58 / (0.91 + 0.58) = 1.0556/1.49 = 0.71

FPR = FP/ (FP+TN) = 18/ (18+5950) = 18/5968 = 0.0030

Similarly the results are calculated for all the NEs and tabulated in Table 10 and Table 11.

Table 9. Total number of NE's in training corpus

Named Entity (NE)	Tag	Tag label	Support
Person	NEP	0	37181
	NEPB	1	8222
	NEPI	2	4338
	NEPE	3	9216
Location	NEL	4	21419
	NELB	5	126
	NELI	6	78
	NELE	7	164
Organization	NEO	8	1090
	NEOB	9	254
	NEOI	10	280
	NEOE	11	202
Designation	NED	12	1495
Term	NETE	13	1065
Title-Person	NETP	14	1587
Title-Object	NETO	15	593
Brand	NEB	16	185
Measurement	NEM	17	1425
Number	NEN	18	946
Time	NETI	19	800
Abbreviation	NEA	20	5218
Noun entity	NE	21	32724
Not a NE	NONE	22	151440

Table 9 indicates the count of different named entitied in the whole training corpus created manually for this work.

Table 10. Classification Results of Test-set Corpus using linear kernel

Named Entity (NE)	Tag	Tag label	Precision	Recall	F1 - score	Support
Person	NEP	0	0.91	0.58	0.71	324
	NEP (BIE)	1 2 3	0.47	0.49	0.48	35
Location	NEL	4	0.93	0.67	0.78	148
	NEL (BIE)	5 6 7	0.86	0.50	0.63	12
Organization	NEO	8	0.72	0.46	0.56	28
	NEO (BIE)	9 10 11	0.50	0.45	0.47	20
Designation	NED	12	0.96	0.61	0.75	44
Term	NETE	13	0.77	0.23	0.35	103
Title-Person	NETP	14	0.65	0.41	0.50	27
Title-Object	NETO	15	0.97	0.34	0.50	99
Brand	NEB	16	1.00	0.22	0.36	9
Measurement	NEM	17	0.83	0.35	0.49	68
Number	NEN	18	0.71	0.36	0.48	151
Time	NETI	19	0.86	0.69	0.77	55
Abbreviation	NEA	20	1.00	0.45	0.62	11
Noun entity	NE	21	0.74	0.62	0.67	397
Not A NE	NONE	22	0.89	0.98	0.93	5469
Average /Total			87%	88%	87.5%	**7000**

It can be seen that the input test sequence is a good mix of all types of possible named entities. We have mixed single word named entities and multiword (beginning, internal and End, BOE) person names, location names and organization names in the corpus. So, it is inferred that the model is well tested for all kinds of possible classification opportunities. Table 10 and Table 11 give the performance of the model indicating the classification ability of the model and the error analysis respectively.

Table 11. Error analysis

Named Entity (NE)	Tag	Tagl abel	FPR	Support
Person	NEP	0	0.0030	324
	NEP (BIE)	1 2 3	0.0031	35
Location	NEL	4	0.0013	148
	NEL (BIE)	5 6 7	0.0002	12
Organization	NEO	8	0.0008	28
	NEO(BIE)	9 10 11	0.0015	20
Designation	NED	12	0.0002	44
Term	NETE	13	0.0011	103
Title-Person	NETP	14	0.0010	27
Title-Object	NETO	15	0.0002	99
Brand	NEB	16	0.0000	9
Measurement	NEM	17	0.0008	68
Number	NEN	18	0.0036	151
Time	NETI	19	0.0010	55
Abbreviation	NEA	20	0.0000	11
Noun entity	NE	21	0.0144	397
Not a NE	NONE	22	0.4509	5469
Average /Total			2.84%	7000

It is interesting that the proposed model works with higher F1-measure 87.5% on a test corpus of 7000 tokens with linear kernel SVM. The time taken for extraction of the features by SVM training model is 1244.594 seconds for a training corpus size of 1, 51,440 tokens. Moreover, it can be seen that the testing time is very less of the order of 2.031 seconds, which mainly depends on the size of test corpus (7,000 words in this case). 10 fold cross validation results of the system in terms of Precision, Recall and F1-measure are 83.7%, 82.5% and 83.1% respectively.

The results of SVM model with different kernels is tabulated in Table 12 for different sizes of Test-Set. It is seen that the SVM model with linear kernel gives highest F1-Score of 87.5% on a test corpus of 7000 tokens

Table 12. F1-scores of SVM model with different kernels

Test-Set size in words	F1-Score in %			
	linear kernel	Poly kernel	Rbf kernel	Sigmoid kernel
7,000	**87.5**	**68.5**	**51**	**51**
11,000	**82**	**51**	**59.7**	**59.7**
15,000	**81**	**51**	**51**	**51**
20,000	**74.5**	**34.4**	**44.6**	**34.4**
30,000	**71.9**	**20.5**	**40.9**	**20.5**
40,000	**70.9**	**12.5**	**32.9**	**12.5**
50,000	**68.4**	**6.6**	**32.5**	**6.6**

7 Conclusion

Natural Language Processing is an important research area containing challenging issues to be investigated. NERC is a class of NLP which is used for extracting named entities from unstructured data. In this context, this paper focuses on NERC in Kannada language, as it is found that little work is done in this area. In this direction, we have conducted an extensive survey in the related area of NLP and based on the survey, we have proposed a problem and the methodology that has been formulated. Various modeling techniques are investigated, out of which design of supervised SVM is reported here. We have developed an efficient model which is trained on a corpus consisting of 1, 51,440 words. From the test corpus, variety of test samples are chosen randomly and fed as input to the SVM model with different kernels. It is interesting to note that the model recognizes the named entities with an average F1-measure of 87.5% and 10 fold cross validation F1-measure of 83.1% for a test corpus of 7000 tokens with linear kernel. The false positive rate of the algorithm is 2.84% for a test corpus of 7000 tokens with linear kernel.

REFERENCES

[1]. Elizabeth D Liddy, 2001. Natural Language Processing. In Encyclopedia of Library and Information Science.2nd edition.

[2]. James Allen, 2007. Natural Language Understanding. Pearson Publication Inc., 2nd edition.

[3]. Kavi Narayana Murthy, 2006. Natural Language Processing. Ess Ess Publications for Sarada Ranganathan Endowment for Library Science, Bangalore, India, 1st edition.

[4]. Gobinda G. Chowdhury, 2003. Natural Language Processing, Annual Review of Information Science and Technology. 37(1):51-89.

[5]. Kashif Riaz, 2010. Rule-based Named Entity Recognition in Urdu. In Proceedings of the 2010 Named Entities Workshop, pages 126-135. Association for Computational Linguistics.

[6]. Khaled Shaalan and Hafsa Raza, 2007. Person Name Entity Recognition for Arabic. In roceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources, pages 17-24. Association for Computational Linguistics.

[7]. Yassine Benajiba, Mona T Diab and Paolo Rosso, 2009. Using Language Independent and Language Specific Features to Enhance Arabic Named Entity Recognition. Int. Arab J. Inf. Technol., 6(5):463-471.

[8]. Padmaja Sharma, Utpal Sharma and Jugal Kalita, 2010. The First Steps towards Assamese Named Entity Recognition. Brisbane Convention Center, 1:1-11.

[9]. Asif Ekbal and Sivaji Bandyopadhyay, 2009. A Conditional Random Field Approach for Named Entity Recognition in Bengali and Hindi. Linguistic Issues in Language Technology, 2(1).

[10]. Asif Ekbal and Sivaji Bandyopadhyay, 2009. Named Entity Recognition in Bengali. A Multi-engine Approach. Northern European Journal of Language Technology, 1(2):26-58.

[11]. Asif Ekbal and Sivaji Bandyopadhyay, 2008. Bengali named entity recognition using support vector machine. In IJCNLP, pages 51-58.

[12]. Maksim Tkachenko, Andrey Simanovsky and St Petersburg, 2012. Named entity recognition: Exploring features. In Proceedings of KONVENS, pages 118-127

[13]. [Ashwini A Shende and Avinash J Agrawa, 2012. Domain specific named entity recognition. Proceedings of the International Conference on Advances in Computer, Electronics and Electrical Engineering, ISBN: 978-981-07-1847-31, doi:10.3850/978-981- 07-1847-3 P0999:484-487.

[14]. David Nadeau, Peter Turney and Stan Matwin, 2006. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. Published at the 19th Canadian Conference on Artificial Intelligence.

[15]. Sujan Kumar Saha, Partha Sarathi Ghosh, Sudeshna Sarkar and Pabitra Mitra, 2008 . Named entity recognition in Hindi using maximum entropy and transliteration. Research journal on Computer Science and Computer Engineering with Applications, pages 33-41.

[16]. Deepti Chopra, Nusrat Jahan and Sudha Morwal, 2012. Hindi named entity recognition by aggregating rule based heuristics and hidden markov model. International Journal of Information, 2(6).

[17]. Sujan Kumar Saha, Shashi Narayan, Sudeshna Sarkar and Pabitra Mitra, 2010. A composite kernel for named entity recognition. Pattern Recognition Letters, 31(12):1591-1597, doi:10.1016/ j.patrec.2010.05.004.

[18]. Sudha Morwal and Nusrat Jahan, 2013. Named entity recognition using hidden markov model (hmm): An experimental result on Hindi, Urdu and Marathi languages. International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), 3(4):671-675.

[19]. Erik F Tjong Kim Sang and Fien De Meulder, 2003. Introduction to the conll-2003 shared task: Languageindependent named entity recognition. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4, pages 142-147. Association for Computational Linguistics.

[20]. Asif Ekbal and Sivaji Bandyopadhyay, 2010. Named entity recognition using support vector machine: A language independent approach. International Journal of Electrical, Computer, and Systems Engineering, 4(2):155-170.

[21]. Kishorjit Nongmeikapam, Tontang Shangkhunem, Ngariyanbam Mayekleima Chanu, Laishram Newton Singh, Bishworjit Salam and Sivaji Bandyopadhyay, 2011. CRF based name entity recognition (ner) in Manipuri: A highly agglutinative Indian language. In Emerging Trends and Applications in Computer Science (NCETACS), 2nd National Conference on, pages 1-6. IEEE.

[22]. Thoudam Doren Singh, Kishorjit Nongmeikapam, Asif Ekbal and Sivaji Bandyopadhyay, 2009. Named entity recognition for Manipuri using support vector machine. In PACLIC, pages 811-818.

[23]. S Pandian, Krishnan Aravind Pavithra and T Geetha, 2008. Hybrid three-stage named entity recognizer for Tamil. INFOS.

[24]. R Vijayakrishna and Sobha Lalitha Devi, 2008. Domain focused named entity recognizer for Tamil using conditional random fields. In IJCNLP, pages 59-66.

[25]. G.V.S.Raju B.Srinivasu, Dr.S.Viswanadha Raju and K.S.M.V.Kumar, 2010. Named entity recognition for Telugu using maximum entropy model. Journal of Theoretical and Applied Information Technology (JATIT), 13:125-130.

[26]. Dr. A. Vinaya Babu, Dr. A. Govardhan, B. Sasidhar and P. M. Yohan, 2011. Named entity recognition in Telugu language using language dependent features and rule based approach. International Journal of Computer Applications (0975-888), 22(8):30-34.

[27]. [S Amarappa, Dr. S V Sathyanarayana, 2013. "A Hybrid approach for Named Entity Recognition, Classification and Extraction (NERCE) in Kannada Documents". Proceedings of the International Conference on Multimedia Processing, Communication and Information Technology (MPCIT-2013). Book Series: Advances in Engineering and Technology Series, IDES publications, DOI: 03.AETS.2013.4.91, ISBN: 2214 - 0344, Volume: 4, Page(s):173-179.

[28]. S Amarappa and S V Sathyanarayana, 2013. Named entity recognition and classification in Kannada language. International Journal of Electronics and Computer Science Engineering, 2(1):281–289.

[29]. S Amarappa, and S V Sathyanarayana, 2015. Kannada Named Entity Recognition and Classification (NERC) based on Multinomial Naïve Bayes (MNB) Classifier. International Journal on Natural Language Computing (IJNLC), DOI: 10.5121/ijnlc.2015.4404, Vol. 4, No.4, Pages 39-52.

[30]. S Amarappa, and. S V Sathyanarayana, 2015. Kannada Named Entity Recognition and classification (NERC) based on Conditional Random fields (CRF). Second International Conference on Emerging Research on Electronics, Computer Science and Technology (ICERECT-2015), PES College of Engineering, Mandya, India. 978-1-4673-9563-2/15/$31.00 ©2015 IEEE.

[31]. David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification, National Research Council, Canada / New York University.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
VOLUME 5, NO. 1
ISSN 2054-7390

Data Cube Representation for patient Diagnosis System Using Fuzzy Object-Oriented Database

Shweta Dwivedi, Dr. Santosh Kumar
Department of Computer Science and Engineerin, Maharishi University of Information Technology, Lucknow, India;
Muit23412@gmail.com; sant7783@hotmail.com

ABSTRACT

In the current scenario, everyone wants to store and fetch the information in an easy and faster way. Therefore, the data cube is one of the leading tools in these days that facilitate the user to store and retrieve the decision support information in a faster manner with ease. In this paper the patient diagnostic system (PDS) is proposed for the patient who is suffered from the several types of fever and modeling of fuzzy object-oriented database. An attempt is made to design a three dimensional data cube for the fuzzy object-oriented database for storing the vague or imprecise information in it. A class, sequence and activity diagrams are also designed for the graphical representation of the proposed work through the well known modeling language i.e. Unified Modeling Language (UML).

Keywords: UML, Activity Diagram, Class Diagram, Fuzzy Object-Oriented Database, Data Cube.

1 Introduction

Modeling is one of the tools to understand the process and flow of input and output of any system. Therefore, the Object Management Group (OMG) has released a well known modeling language i.e. Unified Modeling Language (UML) for designing the huge and complex problems. The word fuzzy defines the vague values or non crisp information; it deals with uncertainty in the information or values which are produced by the human. There are several researchers who produced the data cube for retrieving the desired information within a fraction of time. Let us first describe the previous research done. Saxena et al. [1] have proposed a UML model for the patient registration system and designed a three dimensional data cube for faster searching & storing of patient registration database. Dev and Mishra [2] have presented a decision support in banking sector which link up the strengths of both OLAP and Data Mining for improving the efficiency and to check the emergence & Creation of innovative ways in this field. The DAWA algorithm, standing for a hybrid algorithm of Dct for Data and discrete WAvelet transform, is proposed by Hsieh et al [3] to approximate the cube streams. Li et al. [4] have introduced two techniques called addset data structure and sliding window to deal with this problem. Malvestuto [5] has introduced: (1) a merge operator combining the contents of a primary data cube with the contents of a proxy data cube, (2) merge expressions for general combination schemes, and (3) an equivalence relation between merge expressions having the same pattern. Doka et al. [6] have presented the Brown Dwarf, a distributed system designed to efficiently store, query and update multidimensional data over an unstructured Peer-to-Peer overlay, without the use of any proprietary tool. Morfonios et al. [7] have focused on Relational-

DOI: 10.14738/tmlai.51.2778
Publication Date: 09th March, 2017
URL: http://dx.doi.org/10.14738/tmlai.51.2778

OLAP (ROLAP), following the majority of the efforts so far. We review existing ROLAP methods that implement the data cube and identify six orthogonal parameters/dimensions that characterize them. Zhao et al. [8] have introduced Graph Cube, a new data warehousing model that supports OLAP queries effectively on large multidimensional networks. Chen et al. [9] have showed that OLAP techniques can be performed within a modern DBMS without external servers or the exporting of datasets, using standard SQL queries and UDFs. Roy and Susiu [10] have introduced a principled approach to provide explanations for answers to SQL queries based on intervention: removal of tuples from the databases that significantly affect the query answers. Nandi et al. [11] have detailed real-world challenges in cube materialization and mining tasks on Web-scale datasets. Pacifically identify an important subset of holistic measures and introduce MR-Cube, a MapReduce based framework for efficient cube computation and identification of interesting cube groups on holistic measures. Hung et al. [12] have proposed approximate Greedy algorithms GR, 2GM and 2GMM, which are shown to be both effective and efficient by experiments done on a census data set and a forest-cover-type data set.

2 Experimental Study

2.1 Activity Diagram

A UML activity diagram is designed to represent the process of hospital-based patient diagnostic system step by step which is shown in the fig 1. The diagram depicts that the patient arrived at the hospital's OPD section and filled the registration form after fulfilling the registration eligibility like the patient referred by the doctor, the concerned department and concerned specialist doctor is available otherwise the patient is not registered. The patient treatment file is created and an identity card is issued to the patient for all the further treatments. The patient file is send to the Master of Social Work (MSW) section where the social worker screens the patient file and primarily diagnose the patient and social worker send the primary diagnostic report to the specialist doctor where the doctor diagnoses the patient in details and prescribed the patient to admit in the department which is depending on the condition on the patient. If the doctor recommended the patient to admit in the ward, then the patient is admitted into the concerned word with an allotment of bed and the detailed treatment of the patient is started. If the patient is not recommended by the doctor for admission into the ward the patient take the doctor's prescription and purchase the medicine from the pharmacy and take it as prescribed. After getting well the patient go home from the hospital and the activity diagram is terminated.

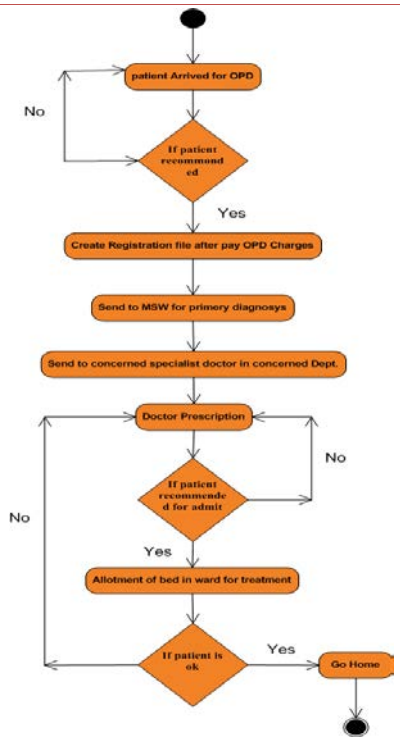

Figure 1. Activity Diagram for Patient Diagnostic System.

2.2 UML Class Diagram

A Complete process of patient diagnostic system is explained in detail through UML class diagram. There are several major classes like Patient, Registration_Desk, Doctor, Departments, Ward and Patient_Discharge represented in the figure 2. The class Patient has single associations with the Registration_Desk and multiple associations with Doctor and Patient_Discharge while the Registration_Desk has multiple associations with the Department class. The Patient_Admit and Patient_Discharge classes have multiple associations with Doctor, Ward and indirectly associated with patient via Doctor Class. The class Ward is further generalizing in Private_Ward and General_Ward.

Therefore, the UML class diagram depicts the complete process of patient diagnostic system in which the patient has arrived at the registration section where the concerned person checks the eligibility (like patient referred by the other doctor, concerned department and doctor is available etc.) of patient for registration. If eligible then the concerned person registers the patient and creates a patient registration file and issued a registration card also. As the patient has registered the MSW sends the patient to the concerned department after primary diagnosis. The patient is detailed diagnosed by the concerned doctor and the patient is admitted in to the ward if the doctor recommended otherwise the doctor prescribed some medicine to the patient and patient go home. The patient who is admitted into the wards goes home after the doctor declared fit.

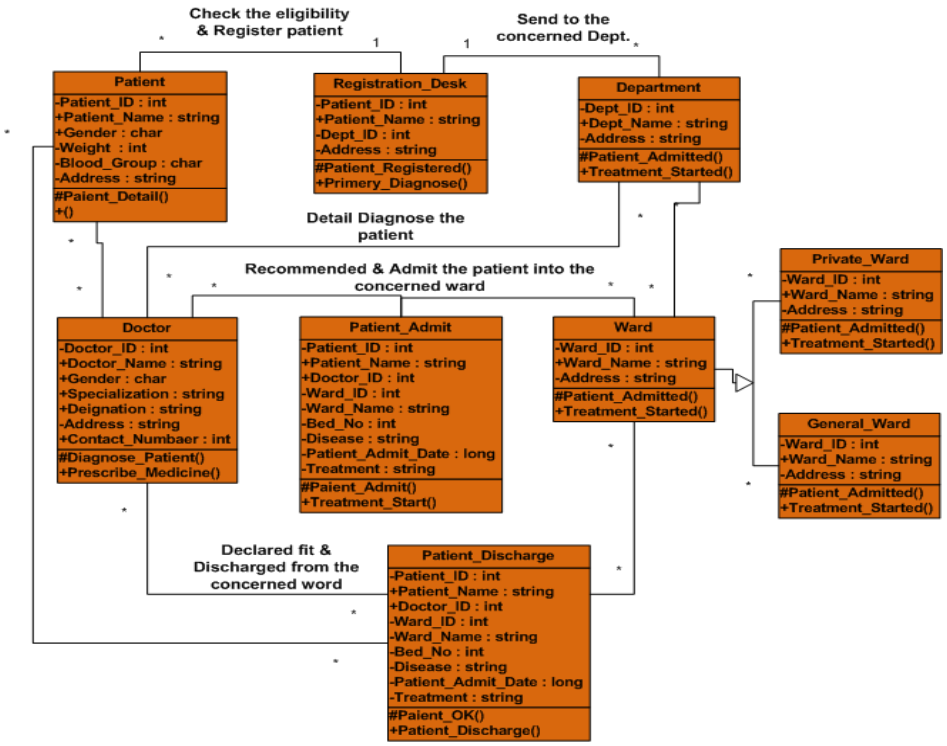

Figure 2. UML Class Diagram for Patient Diagnostic System.

2.3 Fuzzy Object-Oriented Database

Unclear and inconsistent information is handled by the most promising database i.e. the fuzzy database. An extension of the fuzzy database is fuzzy object-oriented database (FOOD) that also deals with the vague or imprecise information as well as it supported the object-oriented programming concepts for storing and interrogating the vague information and turned this vague information into crisp one. Therefore, a fuzzy object-oriented database is designed for the patient diagnostic system (PDS) of **"dengue fever"** with its range value and is represented in the Table 2. Some fuzzy queries are performed, for that the fuzzy query approach is based on the fuzzy logic.

Table 1. Sample Fuzzy Object-Oriented Database for Patient Diagnostic System (PDS).

	P_ID	Name	Gender	Age	Weight	Blood Group	Disease
1	1004	MASTER JAINUL	Male	23	60	A	Dangue
2	1006	BABY RADHA	Male	12	15	O+ve	Dangue
3	1007	MR. JITIN WADHWANI	Male	69	55	AB	Dangue
4	1008	MRS. KIRAN	Female	25	45	AB+	Dangue
5	1011	MRS. NIRMALA	Female	35	46	AB+	Dangue
6	1012	MISS. SHAHEEN BANO	Female	25	45	B+	Dangue
7	1014	MR. RAMESH	Male	37	66	O+	Dangue
8	1016	MR. RAM SWAROOP	Male	30	63	AB+	Dangue
9	1017	MRS. PYARA	Female	17	36	O+	Dangue
10	1018	MR. TULAI GAUTAM	Male	18	38	B+	Dangue
11	1019	MRS. SHAYRA	Female	12	20	AB+	Dangue

2.4 Designing of Data Cube

One of the major tools for decision support system is data cube that represent the desired information or data along with some desired measures. The data cube has some attributes on its each dimension from the database and each cell represents the desired measure values. Many users want to retrieve the desired information they perform several kinds of queries on the data cube and retrieve decision support information.

Let us consider the UML activity diagram to design the three dimensional data cube for the patient diagnostic system (PDS) which contains the three major attributes are considered i.e. Patient_Name lying on (x-axis), Disease lying on (y-axis) and Range_Value (fuzzy value) lying on (z-axis). The axis representation of attributes is shown in the figure 3.

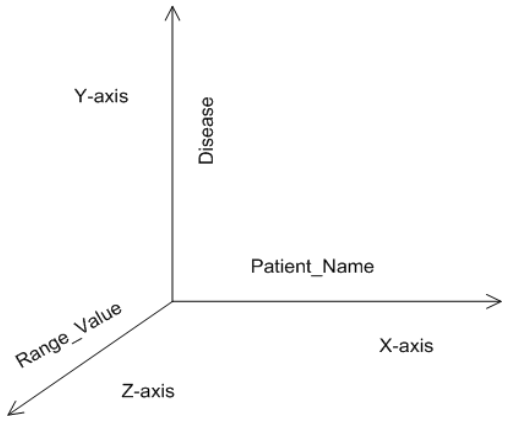

Figure 3. 3-D Axis Representation of Attributes

Therefore, according to the 3-D axis representation of attributes the three domains is taken from the designed fuzzy object-oriented database (FOOD) to design the 3D data cube for the patient diagnostic system (PDS). The database contains the numbers of records of patients which is represented in the table 3.

Table 1. Sample PDS Database with Fuzzy Range_Values

	A	B	C
	P_NAME	DISEASE	MEMBER FUNCTION (RANGE VALUE)
	MR. ANURAG SHARMA	Virul Fever	98.0<101
	MR. JAGDISH	Virul Fever	98.0<101
	MRS. CHANDAWATI	Maleria	98.0<107
	MASTER JAINUL	Dangue	18.0<22.0
	MR. SANTOSH KUMAR	Virul Fever	98.0<101
	BABY RADHA	Dangue	18.0 < 22.0
	MR. JITIN WADHWANI	Dangue	18.0 < 22.0
	MRS. KIRAN	Dangue	18.0 < 22.0
	MRS. USHA MISHRA	Virul Fever	98.0<101
	MISS. SUHASI	Typhoid fever	103<104
	MRS. NIRMALA	Dangue	18.0 < 22.0
	MISS. SHAHEEN BANO	Dangue	18.0 < 22.0
	MISS. ROHINI	Typhoid fever	103<104
	MR. RAMESH	Dangue	18.0 < 22.0
	MASTER ANKIT	Virul Fever	98.0<101
	MR. RAM SWAROOP	Dangue	18.0 < 22.0
	MRS. PYARA	Dangue	18.0 < 22.0
	MR. TULAI GAUTAM	Dangue	18.0 < 22.0
	MRS. SHAYRA	Dangue	18.0 < 22.0
	MR. SHIV DULARE	Virul Fever	98.0<101

Let us now consider the data bank represented into the table 1 for designing the 3-D data cube for patient diagnostic system with fuzzy range values. Each cell of data cube has the combination of three major attributes which are represented into the three domain of the table 1. The sample data cube for the patient diagnostic system who suffered from viral and dengue fever is represented in the figure 4(a) & 4(b) respectively. Each cell represents the attribute values requested by the user. The main objective of designing the data cube is to retrieve the information that can use in making decision in a faster manner. There are some queries are performed to retrieve the necessary information which is shown in the different phases of the data cube.

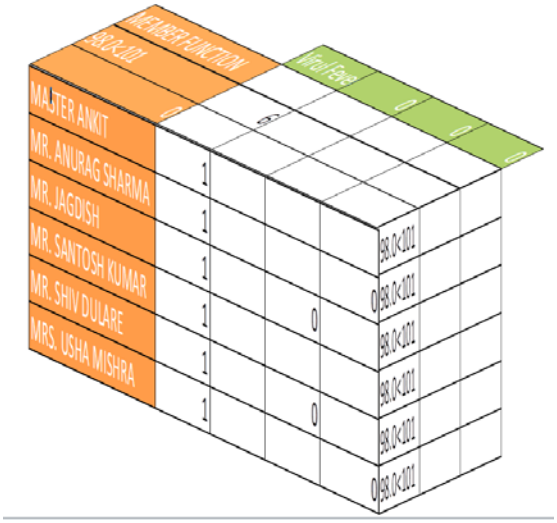

Figure 4(a). Data Cube Shows the Patients Suffering from Viral Fever

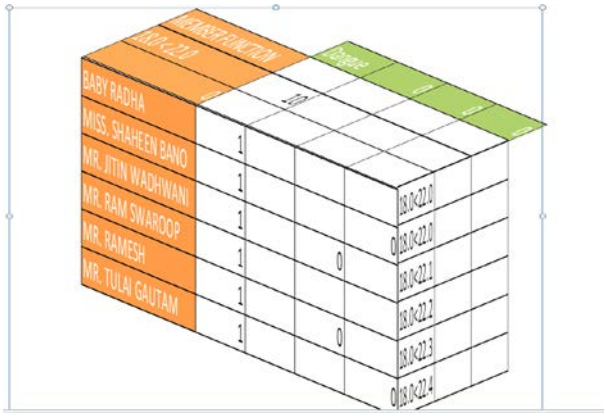

Figure 4(b). Data Cube Shows the Patients Suffering from Dengue Fever

3 Conclusion

There is a large scope to study the various kinds of data cube that provides the facility to the users to access the desired information within the fraction of seconds. Therefore, the present work is an attempt to design a 3D data cube for the designed fuzzy object-oriented database so that one can get the desired information in an easy and faster manner.

ACKNOWLEDGEMENT

Authors are grateful to the Vice-Chancellor (Prof. P.K. Bharti), Maharishi University of Information Technology Lucknow for providing the excellent facility in the computing lab of Maharishi university of Information Technology, Lucknow, India. Thanks are also due to University Grant Commission, India for support to the University.

REFERENCES

[1] Saxena V., Ansari A. G. and Kumar K., Data Cube Representation of Patient Registration System through UML, International Journal of Computer Science and Network Security, Vol. 8, No. 10, October 2008.

[2] Dev H. and Mishra K. S., Design of Data Cubes and Mining for Online Banking System, International Journal of Computer Applications, Vol. 30, No. 3, September 2011.

[3] Hsieh J. M., Chen S. M. and Yu. S. P. 2005. Integrating DCT and DWT for approximating Cube Streams. In Proceedings of the 14th ACM International conference on Information and Knowledge Management (CIKM), pp 179-186.

[4] Li C., Cong G., Tung H. K. A. and Wang S. 2004. Incremental Maintenance of Quotient Cube for Median. In Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data mining, pp 226-235.

[5] Malvestuto M. F., A Join-Link Operator to Combine Data Cubes and Answer Queries from Multiple Data Cubes, ACM Transactions on Database Systems (TODS), Vol. 39, Issue 3, September 2014.

[6] Doka K., Tsoumakos D., and Koziris N. 2010. Distributing the Power of OLAP. In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing (HPDC 10), pp 324-327.

[7] Morfonios K., Konakas S., Ioannidis Y. and Kotsis N., ROLAP implementations of the data cube, ACM Computing Surveys (CSUR), Vol. 39, Issue 4, 2007.

[8] Zhao P., Li X., Xin D., and Han J. 2011. Graph Cube: On Warehousing and OLAP multidimensional Network. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, pp 853-864.

[9] Chen Z., Ordonez C. and Alvarado G. C. 2009. Fast and Dynamic OLAP exploration using UDFs. In Proceedings of the ACM SIGMOD International Conference on Management of Data pp 1087-1090.

[10] Roy S. and Suciu D. 2014. A Formal Approach to Finding Explanations for Database Queries. In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp 1579-1590.

[11] Nandi A., Yu C., Bohannon P., and Ramakrishna R., Data Cube Materialization and Mining over MapReduce, Transactions on Knowledge and Data Engineering, Vol. 6, No. 1, January 2012.

[12] Hung E., Cheung W.D. and KAO B., Optimization in Data Cube System Design, Journal of Intelligent Information Systems, pp 17-45, 2004.