Parsing in Indian Languages

Editors

Kommaluri Vijayanand, MCA
Department of Computer Science
Pondicherry University
Pondicherry - 605014, India
kvpudu@gmail.com

L. Ramamoorthy, Ph.D. Linguistics
Central Institute of Indian Languages
Ministry of Education, Government of India
Mysore 570 006, India
ramamoorthy@ciil.stpmy.soft.net

Foreword

*Parsing in Indian Languages***,** edited by Kommaluri Vijayanand and L. Ramamoorthy, has a key goal of exhibiting and developing the widest promising range of new research in the field of Language Processing as it relates to Indian needs.

Parsing of Indian language texts requires an understanding of the diversity found in the grammatical structures of different languages spoken in India. This may sound to be a complex process, but this can be tackled if we have detailed analysis of natural language structures of these languages. Here is where a deeper and purposeful co-operation between computing experts and linguistics scholars is called for.

In *Parsing in Indian Languages*, academicians, researchers and other Language Processing professionals discuss and present their research and results. They share their discoveries and experiences for further research. This will certainly help impart the latest techniques in the field of Language Processing to the students as well. Ultimately this research will prove useful in many fields including communication.

Collaboration with the Central Institute of Indian Languages has opened ways to understand, describe and explain the varied structures actually used in major Indian languages. This volume is a product of this collaborative effort of the Department of Computer Science, Pondicherry University with the Central Institute of Indian Languages, Mysore.

This volume is an excellent source for the scientific community interested in Language Processing . I hope that our collaborative effort will bring out many more such research proceedings in the future as well.

We at the Department of Computer Science, Pondicherry University are committed to carry forward this work through excellent research based activities in a collaborative fashion. This collaborative platform will serve as a new medium for the scientific and research community, centers and agencies including the participating nodes.

Dr. P. Dhavachelvan
Professor and Head
Department of Computer Science
Pondicherry University
Puducherry - 605014

Preface

Ancient Indian grammarians excelled themselves in identifying the components of sentences and grouping them in terms of their structural meaning/s as well as contextual meaning/s in utterances. Their practices assumed certain language universals across human languages. At the same time, they were always aware of the specific features of individual languages. Sanskrit grammatical traditions influenced the development of grammars in several languages such as Telugu, Kannada and Malayalam. The Tamil ancient tradition followed its own course in most areas and developed its own descriptive technical terms and language-specific features of the Tamil language. However, in both the traditions, parsing played a very important role.

Parsing stands for the processes of analysis applied on sentences to identify its constituents. Generally speaking, parsing may be described as the process of identifying the parts of speech, analysis of sentence and sentence types and their constituents. Universal as well as language-specific features may be identified through parsing. An important requirement, often ignored after the introduction of the teaching of English grammar in Indian schools, is the emphasis on language-specific features that are often context-sensitive as well as based on semantic and lexical relations peculiar to a language. English school grammar began to dominate the thinking of the educated people in India. Native grammar traditions also focused more on the earlier stages of the language as exemplified in traditionally respected grammar books.

Modern structural linguistics enabled us to overcome these two limitations and to look at linguistic structures from descriptive and distributional points of views current usage. Further developments within linguistics such as the emergence of the generative grammar models, etc. along with the emergence of computer science and programming, have given us new insights into the processes and models of parsing. In addition, we now recognize that in order to take the benefits of computing, to a variety of languages, we need closer scrutiny and formalization of parsing of target languages.

India has many languages and a good number of these may be termed as major languages in terms of the expansive nature of their use in various fields. Use and development of a variety programmes for the efficient use of computer and computing in these languages will be better achieved if we do the parsing of the syntactic and semantic structures of these languages using well developed concepts and practices of parsing dealing with the universal and specific features of these languages.

The present volume of papers is an attempt to take up some important problems in the field of parsing and apply these techniques to some Indian languages. We do believe that this volume will help both teachers and students of Computer Science courses in Indian Universities. These papers will help Indian researchers and software engineers to continue to identify specific features of Indian languages and find solutions to solve them.

These papers were presented in the National Seminar on Lexical Resources and Computational Techniques on Indian Languages organized by the Department of Computer Science during 04th and 05th October 2010 in Pondicherry University.

Our grateful thanks are due to the following:

Prof. J.A.K. Tareen, Vice-Chancellor, Pondicherry University for his consistent support and encouragement extended to us for hosting this National Seminar on Lexical Resources and Computational Techniques on Indian Languages on the campus. Our gratitude is extended to our sponsors The Department of Information Technology under the Ministry of Information and Communications Technology, Government of India, Central Institute of Indian Languages, Mysore and Department of Science, Technology and Environment, Government of Puducherry. Prof. Rajesh Sachdeva, Director, Central Institute of Indian Languages, Mysore had inaugurated the Seminar and motivated the research community with his inspirational inaugural address stressing on mobilizing young scholars to become part of the mission of translation.
We are thankful to Prof. Pushpak Bhattacharya, Department of Computer Science, Indian Institute of Technology-Bombay, Mumbai is instrumental in motivating the young research scholars towards contributing one's resources to the translation mission.

We extend our gratitude to Prof. Rajeev Sangal, Director, Indian Institute of Information Technology, Hyderabad, Prof. V. Prithiviraj, Dean (i/c), School of Engineering and Technology, Pondicherry University, Prof. R. Subramanian, Prof. Kavi Narayana Murthy, Prof. Panchanan Mohanty, Prof. G. Umamaheswara Rao of University of Hyderabad and Thiru. S. Loganathan, Registrar, Thiru. S. Raghavan, Finance Officer of Pondicherry University.

Vijayanand Kommaluri, M.C.A.
L. Ramamoorthy, Ph.D.

Contents

Statistical Machine Translation using Joshua:
An approach to build "enTel" system

[+]Anitha Nalluri,*Vijayanand Kommaluri
+Advisory System Analyst, IBM India, Bangalore, India.
**Assistant Professor*, Dept of Computer Science, Pondicherry University, India
Email:anallur@in.ibm.com

1.0 Abstract

This paper addresses an approach to build "enTel" System – An English to Telugu Machine Translation (MT) System using Statistical Machine Translation (SMT) techniques and Johns Hopkins University Open Source Architecture (JOSHUA). It provides a heuristic approach - To train a probabilistic alignment model and use its predictions to align words and ensure the well form of the target language sentences - The tuning of weights of model to balance the contribution of each of the component parts to find the optimal weights among different models – Evaluation of the quality of machine translation with the Bilingual Evaluation Understudy (BLEU) that compares a system's output against reference human translations.

2.0 Introduction

Machine translation (MT), also known as "automatic translation" or "mechanical translation," is the name for computerized methods that automate all or part of the process of translating from one human language to another. Languages are challenging, because natural languages are highly complex, many words have various meanings and different possible translations, sentences might have various readings, and the relationships between linguistic entities are often vague. The major issues in MT involve ambiguity, structural differences between languages, and multiword units such as collocations and idioms. If sentences and words only had one interpretable meaning, the problem of interlingual translation would be much easier. However, languages can present ambiguity on several levels. If a word can have more than one meaning, it is classified as lexically ambiguous. An approach to solving this problem is statistical analysis.

3.0 Statistical Machine Translation

Statistical Machine Translation (SMT) is founded on the theory that every source language segment has any number of possible translations, and the most appropriate is the translation that is assigned the highest probability by the

system. It requires a bilingual corpus for each language pair, a monolingual corpus for each target language, a language modeler and a decoder. A language model analyses the monolingual TL corpus in order to 'learn' a sense of grammaticality (e.g. word order), based on n-gram statistics (usually trigrams), and then calculates the probabilities of word x following word y etc. in the TL. The probabilities are calculated during the preparation stage and stored. When presented with a new translation, the SL segments are segmented into smaller phrases. They are matched with source language equivalents in the corpus and their translations harvested by the decoder. As the search space is theoretically infinite, the decoder uses a heuristic search algorithm to harvest and select appropriate translations. The translation problem can be describes as modeling the probability distribution Pr(E|T) Where E is the string in Source language and T is the string in Target Language.

$$\Pr(E|T) = \frac{\Pr(T|E)\Pr(E)}{\Pr(T)}$$

Where, Pr(E) is called Language Model (LM) and Pr(T|E) is called Translation Model (TM).

The use of statistical techniques in machine translation has led to dramatic improvements in the quality of research systems in recent years. The statistical machine translation is rapidly progressing, and the quality of systems is getting better and better. An important factor in these improvements is definitely the availability of large amounts of data for training statistical models. Yet the modeling, training, and search methods have also improved since the field of statistical machine translation was pioneered by IBM in the late 1980s and early 1990s.

3.1 N-GRAM Modeling

An n-gram is a subsequence of n items from a given sequence. The items in question can be phonemes, syllables, letters, words or base pairs according to the application. An n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram" (or, less commonly, a "digram"); size 3 is a "trigram"; and size 4 or more is simply called an "n-gram". Some language models built from n-grams are "(n − 1)-order Markov models". An n-gram model is a type of probabilistic model for predicting the next item in such a sequence. N-gram models are used in various areas of statistical natural language processing and genetic sequence analysis. The n-gram model, a special type of a Markov model, predicts the occurrence of the ith word vi with the formula:

$P(v_i) = [\ c(v_i – (n\text{-}1) \ldots v_i)\ ] / [c(v_i – (n\text{-}1) \ldots v_{i\text{-}1})]$

In this formula, c(x) is the number of occurrences of event x. The most significant results in SBMT have been achieved using n-gram modeling and the most common approach is the trigram model, where n = 3.

3.2 SRILM

SRILM is a collection of C++ libraries, executable programs, and helper scripts designed to allow both production of and experimentation with statistical language models for speech recognition and other applications. The toolkit supports creation and evaluation of a variety of language model types based on N-gram statistics, as well as several related tasks, such as statistical tagging and manipulation of N-best lists and word lattices.

3.3 GIZA++

GIZA++ is the Statistical Machine Translation toolkit which was developed by Statistical Machine Translation Team during the summer workshop in 1999 at the Center for Language and Speech Processing at Johns-Hopkins University (CLSP/JHU). It is an extension of the program GIZA (part of the SMT toolkit EGYPT). GIZA ++ is used to train the IBM models 1-5 and HMM Word Alignment model and various smoothing techniques for fertility, distortion/alignment parameters. The training of the fertility models is significantly more efficient.

3.4 Bilingual Evaluation Understudy

The primary programming task for a Bilingual Evaluation Understudy (BLEU) is to compare n-grams of the candidate with the n-grams of the reference translation and count the number of matches. These matches are position independent. The more the matches, the better the candidate translation is. BLEU's strength is that it correlates highly with human judgments by averaging out individual sentence judgment errors over a test corpus rather than attempting to divine the exact human judgment for every sentence: quantity leads to quality[1]. Thus the BLEU method is used for evaluation of quality of machine translation systems.

4.0 Overview of JOSHUA Architecture

Joshua is an open-source toolkit for parsing-based machine translation that is written in Java. JOSHUA decoder assumes a probabilistic synchronous context-free grammar (SCFG) [2]. During decoding, each time a rule is called to construct a new constituent, a number of feature functions are called in order to give a cost for that constituent.

4.1 Translation Grammars

There are a series of classes which define how grammars are created and used. Initially a Grammar Factory to be

constructed to handle the intricacies of parsing grammar files in order to produce a Grammar. This separation is used to decouple the file format from the in-memory representation with the same data structure but different file parsers. The Grammar mostly serves as a wrapper around TrieGrammar in order to give a holistic object representing the entire grammar, though it also gives a place to store global state which would be inappropriate to store in each TrieGrammar object. The TrieGrammar implements a trie-like interface for representing dotted rules for use in parsing charts. This abstract trie can also be viewed as an automaton. Each state of the automaton is represented by a TrieGrammar object [3].

RuleCollection is a collection of individual Rule objects. If these states of TrieGrammar objects are "final" then there is a RuleCollection which could be applied at the current position in parsing. This RuleCollection gives the candidate set of rules which could be applied for the next step of the chart-parsing algorithm. Each of these rules is passed to the PhraseModelFF feature function which will produce the cost for applying that rule [3].

4.2 Convolution

Sometimes for simple implementations this detailed separation of GrammarFactory, Grammar, TrieGrammar, and RuleCollection may seem like overkill. An important thing to keep in mind is that since these are all interfaces, a given implementation can have a smaller number of classes which implement more than one interface [3].

4.3 Language Models

Similarly there are a number of classes that play into language modeling. The NGramLanguageModel interface defines what it means to be a language model. An object of that type is given to the LanguageModelFF feature function which handles all the dynamic programming and N-gram state maintenance [3].

4.4 Minimum Error Rate Training

To balance the contribution of each of the component parts (language model probability, translation model probabilities, lexical translation probability, etc) of the model, the weights should tune to run Minimum Error Rate Training (MERT) for finding the optimal weights among different models [3].

4.5 Evaluation of Translation Quality

The quality of machine translation is commonly measured using the BLEU metric, which automatically compares a system's output against reference human

translations. The BLUE metric can be computed using built-in function of "JoshuaEval" [3]. The translation quality can be further improved by varying the size and weights of training data.

5.0 Machine Translation Systems – Telugu Language – Scenario

Telugu is classified as a Dravidian language with heavy Indo-Aryan influence spoken in the Indian state of Andhra Pradesh. Telugu has the third largest number of native speakers in India (74 million according to the 2001 census) and is 15th in the Ethnologue list of most-spoken languages worldwide.

Sampark – Machine Translation among Indian Languages developed by the consortium of 11 Indian institutions led by International Institute of Information Technology-Hyderabad (IIIT-H) is slated for national launch [4]. It can also translate entire webpage with pictures and graphics intact. Anusaaraka - A machine Translation system has been built from Telugu, Kannada, Bengali, Punjabi and Marathi to Hindi [5]. It is domain free but the system has been applied mainly for translating children's stories. Anubharti - A machine-aided-translation is a hybridized example-based machine translation approach that is a combination of example-based, corpus-based approaches and some elementary grammatical analysis. The example-based approaches follow human-learning process for storing knowledge from past experiences to use it in future [6]. AnuBharti II - the traditional EBMT approach has been modified to reduce the requirement of a large example-base. This is done primarily by generalizing the constituents and replacing them with abstracted form from the raw examples. Matching of the input sentence with abstracted examples is done based on the syntactic category and semantic tags of the source language structure[7].

6.0 Development of "enTel" System

An "enTel" system using Joshua is developed and piloted to find the feasibility and effectiveness of statistical machine translation system between English- Telugu languages. A parallel corpus of south Asian languages called Enabling Minority Language Engineering (EMILLE) for Telugu Language developed by the Central Institute for Indian Languages, Mysore, India and "English to Telugu Dictionary" developed by Charles Philip Brown is considered for training of datasets. The language model is trained using SRILM and GIZA++ tools. The size and weights of training data are tuned to achieve the better quality of machine translation system. The quality of the machine translation system is assessed using BLUE metric.

7.0 Conclusion

The piloted "enTel" System is observed to be an efficient and feasible solution of open MT system for English to Telugu. The "enTel" system requires more enormous amounts of parallel text in the source and target text to achieve high quality translation. SMT gives better results as more and more training data is available. The future work of enTel system is proposed to develop the user interfaces that can retrieve the translated text from source language to targeted language with an ease of clicking a mouse.

8.0 References

[1] *Kishore Papineni etal., (2002),* "BLEU: a Method for Automatic Evaluation of Machine Translation", *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 311-318.*

[2] *Zhifei Li etal., (2009*), "Joshua: An Open Source Toolkit for Parsing-based Machine Translation", *Proceedings of the Fourth Workshop on Statistical Machine Translation , pages 135–139, Athens, Greece, 30 March – 31 March 2009.*

[3] http://www.clsp.jhu.edu/wiki2/Joshua_architecture, *Site last visited 2nd October 2010.*

[4] http://syedakbarindia.blogspot.com/2010/08/iiit-hyderabad-develops-machine.html, *Site last visited 2nd October 2010.*

[5] *Rajeev Sangal etal., (1997)* "ANUSAARAKA: Machine Translation in Stages" , *Appeared in Vivek - A Quarterly in Artificial Intelligence, Vol.10, No.3 (July 1997), NCST, Mumbai, pp.22-25.*

[6] *Renu Jain etal., (2001),* "ANUBHARTI: Using Hybrid Example-Based Approach for Machine Translation", Proc. Symposium on Translation Support Systems (STRANS2001), February 15-17, 2001, Kanpur, India.

[7] *R.M.K. Sinha (2004)* "An Engineering Perspective of Machine Translation: AnglaBharti-II and AnuBharti-II Architectures", *Proceedings of International Symposium on Machine Translation, NLP and Translation Support System (iSTRANS- 2004), November 17-19, 2004, Tata Mc Graw Hill, New Delhi.*

Layered Parts of Speech Tagging for Bangla

Debasri Chakrabarti

CDAC, Pune

debasri.chakrabarti@gmail.com

Abstract-In Natural Language Processing, Parts-of-Speech tagging plays a vital role in text processing for any sort of language processing and understanding by machine. This paper proposes a rule based Parts-of-Speech tagger for Bangla with layered tagging. There are 4 levels of Tagging which also handles the tagging of Multi verb expressions.

I. Introduction

The significance of large annotated corpora is a widely known fact. It is an important tool for researchers in Machine Translation (MT), Information Retrieval (IR), Speech Processing and other related areas of Natural Language Processing (NLP). Parts-of-Speech (POS) tagging is the task of assigning each word in a sentence with its appropriate syntactic category called Parts-of-Speech. Annotated corpora are available for languages across the world, but the scenario for Indian languages is not the same.

In this paper I have discussed a rule based POS tagger for Bangla with different layer of tagging. The paper also shows how the layered tagging could help in achieving higher accuracy.

The rest of the paper is organized in the following way- Section 2 gives a brief overview of Bangla and the process of tagging with examples, Section 3 discusses layered POS Tagging and section 4 concludes the paper.

II. POS Tagging in Bangla

Bangla belongs to Eastern Indo-Aryan group, mainly spoken in West Bengal, parts of Tripura and Assam and Bangladesh. Bangla is the official language of West Bengal and Tripura and the national language of Bangladesh. It is a morphologically rich language, having a well-defined classifier system and at times show partial agglutination. In this section I propose a rule-based POS tagging for Bangla using context and morphological cue. The tag set are both from the common tag set for Indian Languages (Bhaskaran et al.) and IIIT Tag set guidelines (Akshar Bharti). For the top level following tags are taken as given in Table 1. This includes the 12 categories that are identified as the universal categories for the Indian languages from the common tag set framework.

Table 1

Top Level Tagging

	TAGSET	DESCRIPTION
1.	NN	Noun
2.	NNP	Proper Noun
3.	NUM	Number
4.	PRP	Pronoun
5.	VF	Verb finite
6.	VB	Verb Base
7.	VNF	Verb Nonfinite
8.	JJ	Adjective
9.	QF	Quantifier
10.	RB	Adverb
11.	PSP	Postposition
12.	PT	Particle
13.	NEG	Negative
14.	CC	Coordinating
15.	UH	Interjection
16.	UNK	unknown
17.	SYM	Symbol

After the top level annotation there is a second level of tagging. The tag sets are shown in Table 2.

Table 2

Second Level Tagging

	TAGSET	DESCRIPTION
1.	CM	Casemarker
2.	CL	Classifier
3.	CD	Cardinal
4.	CP	Complementizer
5.	DET	Determiner
6.	INTF	Intensifiers
7.	QW	Question Word
8.	SC	Subordinating Conjunction

A. Approaches to POS Tagging

POS tagging is typically achieved by rule-based systems, probabilistic data-driven systems, neural network systems or hybrid systems. For languages like English or French, hybrid taggers have been able to achieve success percentages above 98%. [Schulze et al, 1994]. The works available on Bangla POS Tagging are basically statistical based- Hidden Markov Model (HMM) [Ekbal et al.], Conditional Random Field (CRF) [Ekbal et al.], Maximum Entropy Model [Dandapat]. In this paper we talk about a Rule Based POS Tagger for Bangla. The aim is to proceed towards a hybrid POS Tagger for the language in future.

B. Steps to POS Tagging

The first step towards POS tagging is morphological analysis of the words. For this a Noun Analysis and a Verb Analysis had been done. Nouns are divided into three paradigms according to their endings, these three paradigms are further classified into two groups depending on the feature ± animate. The suffixes are then classified based on number, postposition and classifier information. Verbs are classified into 6 paradigms based on morphosyntactic alternation of the root. The suffixes are further analysed for person and honourofic information. Noun Analysis is shown in Table 1 and Verb Analysis is shown in Table 3.

Table 3

Noun Paradigm

Paradigm	No	Animate	Honourofic	Del Char	Classifier	Case	Form
chele 'boy'	Sg	+	+	0	-	Direct	chele 'boy'
chele 'boy'	Sg	+	+	0	Ti	Oblique	cheleTi 'boy'
chele 'boy'	PL	+	+	0	rA	Direct	cheleraa 'boys'
chele 'boy'	PL	+	+	0	der	Oblique	cheleder 'boys'
chele 'boy'	PL	+	-	0	gulo	Oblique	chelegulo 'boys'
phuul 'flower'	Sg	-	-	0	-	Direct	phuul 'flower'
phuul 'flower'	Sg	-	-	0	TA	Oblique	phuulTA 'flower'
phuul 'flower'	Sg	-	-	0	Ti	Oblique	phuulTi 'flower'
phuul 'flower'	PL	-	-	0	gulo	Direct	phuulgulo 'flowers'
phuul 'flower'	PL	-	-	0	gulo	Oblique	phuulgulo 'flowers'

Verb analysis based on Tense, Aspect, Modality, Person and Honouroficity (TAMPH) matrix is shown in Table 4.

Table 4

Verb Paradigm

Tense	Asp	Mod	Per	Hon	Eg.
Present	fct	-	1st	-	kor-i 'I do'
Present	fct	-	2nd	-	kar-o 'You do'
Present	fct	-	2nd	+	kar-un 'You (Hon) do'
Present	fct	-	3rd	-	kar-e 'He does'
Present	fct	-	3rd	+	kar-en 'He (Hon) does'
Past	Inf	-	2nd	-	kar-ar chilo 'was to be done'
Future	-	-	3rd	+	kor-be-n 'He (Hon) will do'
Present	Dur	-	3rd	-	kor-che 'He is doing'
Present	fct	Abl	3rd	-	kor-te pare 'He can do'

Based on this analysis a MA will return the following for the sentence *'ekjon chele boigulo diyeche'*

1. *ekjon (NN,CD) chele (NN) boigulo (NN) diyeche (VF)* 'A boy gave the books'

These are the simple tags that a MA can give. To reduce the ambiguity we need linguistic rules. The ambiguity here is between a Cardinal and a Noun. *ekjon* 'one' can be both- a Noun and a Cardinal. To resolve this sort of ambiguity following rule is given

> Noun vs. Cardinal: if the following word is a noun without a suffix and the token to be processed can qualify the succeeding noun, then the processing token is a cardinal, otherwise it is a noun. [eg. in *ekjon chele*, *ekjon* can be a cardinal or noun, but as it can qualify *chele*, and *chele* is without a suffix it will be an cardinal, not a noun]

The POS tagger will go through 3 stages. At the first stage preliminary tags will be assigned with the help of MA and disambiguating rules. Stage 2 will do a deeper level analysis and provide information like Classifier, TAMPH, Postposition etc. Stage 3 or final stage will run a local word grouper and give the noun group and verb group information. Fig.1. shows stage by stage output of the POS Tagger of the sentence *ekTi shundori meye nodir dhare daNRiye ache* 'One beautiful girl is standing on the bank of the river'

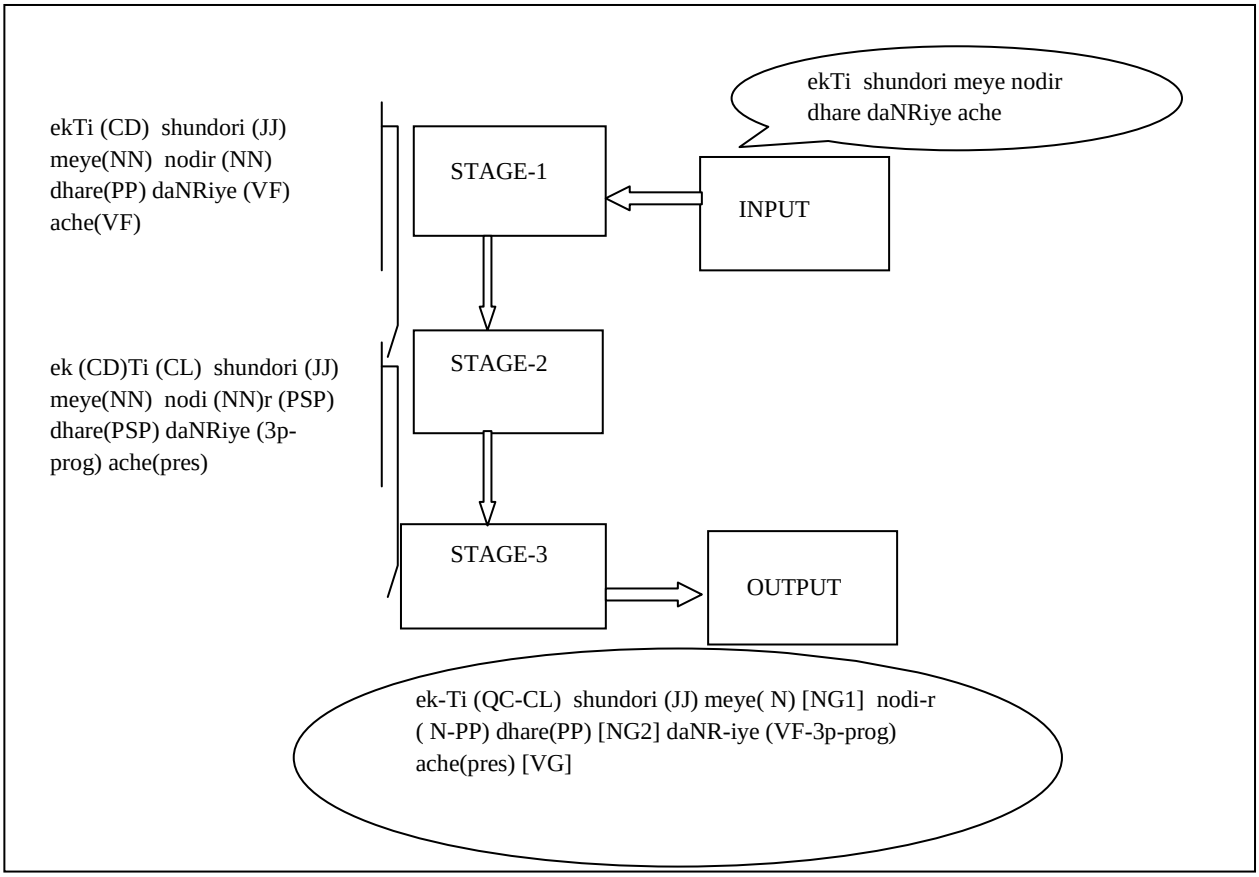


Fig. 1. Stages of POS Tagger

III. Handling Multi Verb Expressions

The POS Tagging process described in this paper till now will be able to tag and group simple verbs. Multi verb expressions (MVE) are not taken care here. MVEs are very frequent in South Asian Languages. These MVEs can be of two types-

a. Noun+Verb Combination, e.g., aarambha karaa 'to start'

b. Verb+Verb Combination e..g., kore phæla 'to do'

The former type of constructions is commonly known as Conjunct Verbs while the latter is called Compound Verb. The Tag set explained here does not include tags for this sort of combination. Therefore, examples like 2 and 3 will have the following tagging-

2. chelegulo kaajTaa aarambha koreche 'The boys started the work'

NN	NN	NN	VF
NN-CL	NN-CL	NN	VF-3p-pt.
[NG1]	[NG2]	[NG3]	[VG]

3. kaajTaa bandho hoyeche 'The work stopped'

NN	NN	VF
NN-CL	NN	VF-3p-pt.
[NG1]	[NG2]	[VG]

Both in 2 and 3 *aarambha koreche* 'started' and bandho *hoyeche* 'stopped' are instances of conjunct verbs. The information of conjunct verb is missing from the tagged output which is leading to a wrong verb group and Noun group identification. As of now both *aarambha* 'start' and bandho 'stop' are considered as Nouns and *koreche* 'do' and *hoyeche* 'happen' as verbs. Due to this the local word grouper

has grouped both *aarambha* 'start' and bandho 'stop' as [NG]. This will lead to wrong syntax affecting the accuracy of the system. To handle this sort of situation I suggest here to add one more layer of tagging before word grouping. The third level of tagging is shown in Table 5.

Table5.

Third Level Tagging

	TAGSET	DESCRIPTION
1.	CNJV	Conjunct Verb
2.	CPDV	Compound Verb

Fig. 2. shows the modified stage by stage output of the POS Tagger of the sentence *chelegulo kaajTaa aarambha koreche* 'The boys started the work'

IV. Conclusion and Future Work

In this paper I have discussed a rule based POS tagger for Bangla with layered tagging. There are four levels of Tagging. In the first level ambiguous basic category of a word is assigned. Disambiguation rules are applied in the second level with more detail morphological information. At the third level multi word verbs are tagged and the fourth or the final level is the level of local word grouping or chunking.

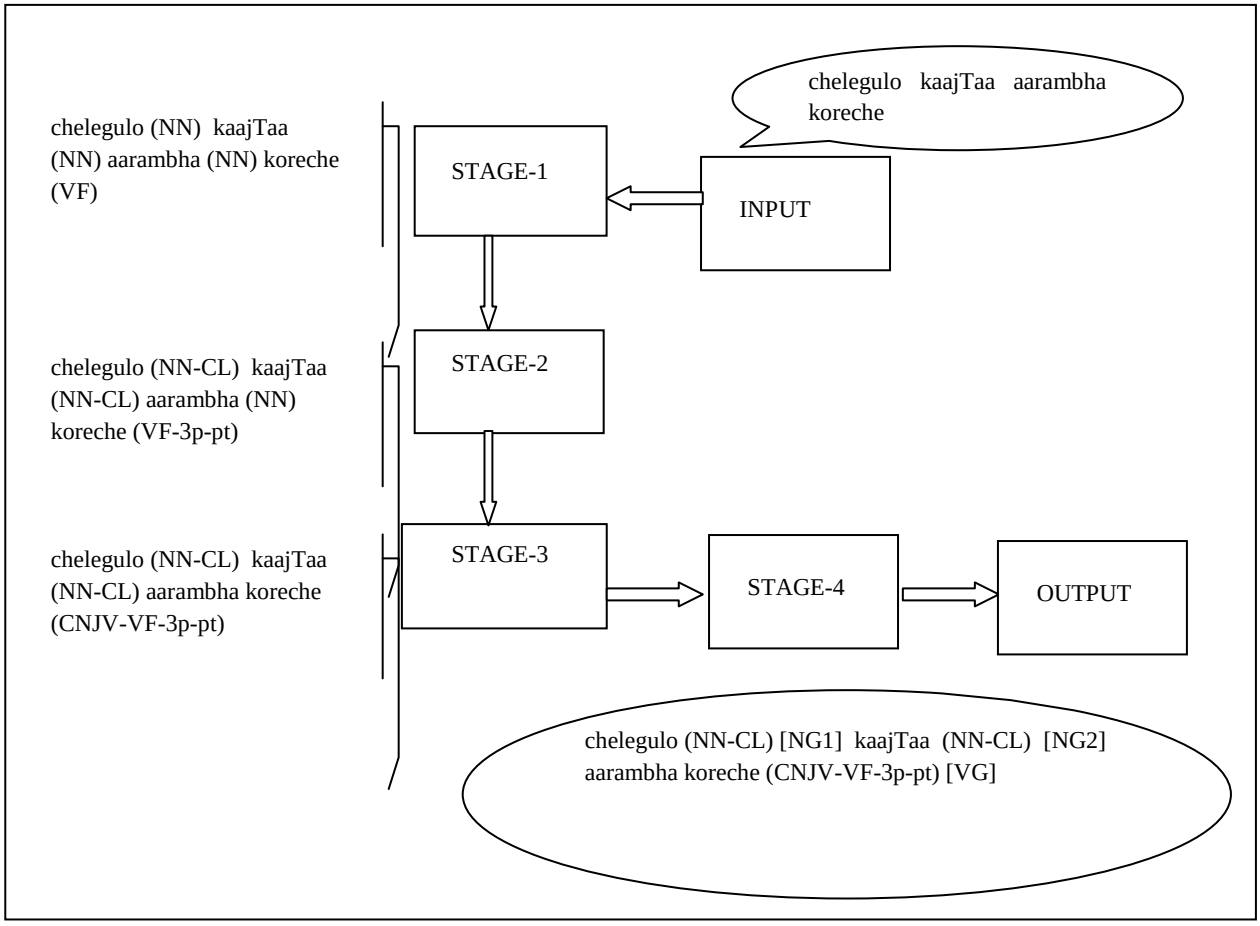


Fig. 2. Modified Stages of POS Tagger

REFERENCES

[1] Akshar Bharati, Rajeev Sangal, Dipti Misra Sharma and Lakshmi Bai. 2006. AnnCorra:*Annotating Corpora Guidelines for POS and Chunk Annotation for Indian Languages*, Technical Report, Language Technologies Research Centre IIIT, Hyderabad.

[2] ARONOFF, MARK. 1976. Word Formation in Generative Grammar. Cambridge: MA: MIT Press
SINCLAIR, J. 1991. Corpus, concordance, collocation. Tuscan Word Centre, Oxford: Oxford University Press

[3] ARONOFF, MARK. 2004. Developing Linguistic Corpora: A Guide to good practice. Oxford: Oxford University Press

[4] Banko, M., & Robert Moore, R. Part of speech tagging in context. 20th International Conference on Computational Linguistics. 2004

[5] Baskaran S. et al. Designing a Common POS-Tagset Framework for Indian Language. The 6th Workshop on Asian Language Resources. 2008

[6] Dandapat, S. Part-of-Speech Tagging and Chunking with Maximum Entropy Model. Workshop on Shallow Parsing for South Asian Languages. 2007.

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

[8] Debasri Chakrabarti, Vaijayanthi M Sarma, Pushpak Bhattacharyya. Compound Verbs and their Automatic Extraction 22nd International Conference on Computational Linguistics, Manchester. 2008

[9] Debasri Chakrabarti, Vaijayanthi M Sarma, Pushpak Bhattacharyya. Identifying Compoud Verbs in Hindi. South Asian Language Analysis. 2006

[10] Ekbal, A., Mandal, S., & Bandyopadhyay, S. POS tagging using HMM and rule based chunking . Workshop on Shallow Parsing for South Asian Languages. 2007.

[11] IIIT-tagset. A Parts-of-Speech tagset for Indian languages. http://shiva.iiit.ac.in/SPSAL2007/iiit_tagset_guidelines.

[12] Saha, G.K., Saha, A.B., & Debnath, S. Computer Assisted Bangla Words POS Tagging. Proc. International Symposium on Machine Translation NLP & TSS. 2004.

[13] Soma Paul. An HPSG Account of Bangla Compound Verbs with LKB Implementation, A Dissertation, CALT, University of Hyderabad, 2004.

[14] Smriti Singh, Kuhoo Gupta, Manish Shrivastava, and Pushpak Bhattacharyya. 2006. Morphological richness offsets resource demand – experiences in constructing a pos tagger for hindi In Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 779–786, Sydney, Australia, July. Association for Computational Linguistics.

DEVELOPING MORPHOLOGICAL ANALYZERS FOR FOUR INDIAN LANGUAGES USING A RULE BASED AFFIX STRIPPING APPROACH

Mona Parakh
Reader/Research Officer
ldc-monaparakh@ciil.stpmy.soft.net

Rajesha N
Senior Technical Officer,
ldc-rajesha@ciil.stpy.soft.net

Linguistic Data Consortium for Indian Languages, CIIL, Mysore

Abstract **- The present paper deals with the design and development of morphological analyzers for four Indian languages, viz., Assamese, Bengali, Bodo and Oriya. These analyzers are being developed using the Suffix Stripping Approach.**

The results of the first version of the analyzers using this approach are fairly encouraging. The coverage of the system is directly related to the size of the dictionary. As this is an ongoing work, we hope to expand and make the system more robust, by increasing the dictionary size.

I. INTRODUCTION

Considering the extensive work that is being carried out in the area of Indian Language Technologies, towards building Language Applications for Major Indian Languages it is the need of the hour to develop and generate language resources for a large number of Indian languages, which are of high quality and with distinct standards.

In order to fulfill this long-pending need, the Central Institute of Indian Languages, Mysore and several other institutions working on Indian Languages technology have set up the Linguistic Data Consortium for Indian Languages (LDC-IL), whose main goal is to create and manage large Indian languages databases. One of the many resource building activities that LDC-IL is involved in includes developing Morphological Analyzers and Generators for major Indian languages.

There are two approaches used to build the Morphological Analyzers at LDC-IL, viz., the Word and Paradigm Approach [1] and the Rule Based Affix Stripping Approach. Morphological Analyzers for ten of the thirteen Indian languages mentioned above are being developed using the Apertium – Lttoolbox [2]. and [5]. For four of the languages, viz., Assamese, Bengali, Bodo and Oriya, analyzers are being developed using the suffix stripping approach. Some other research groups have developed analyzers using the Apertium-Lttoolbox for languages like Marathi [6], Telugu and Tamil [3].

The present paper reports the ongoing work of building Morphological Analyzers using the Suffix Stripping method for the four languages – Assamese, Bengali, Bodo and Oriya. Currently the system only handles inflectional suffixes though it will be further modified so as to handle derivation as well as prefixation, in each of these languages. The system is at different stages of completion depending on the availability of the language resources and human resources for the respective languages.

II. RULE BASED SUFFIX STRIPPING APPROACH.

The Word and Paradigm Model (WPM) is unsuitable and inadequate to capture all morphological functions in case of Assamese, Bengali, Bodo and Oriya. The reason for this is that these languages are classifier based languages. Even though the classifiers are finite in number, they can occur in various combinations with nouns. This would increase the manual effort of paradigm creation immensely. Moreover, in these languages morpho-phonemics does not play much of a role. Hence, the Suffix Stripping Approach has been found to be suitable.

As the name suggests, this method involves identifying individual suffixes from a series of suffixes attached to a stem/root, using morpheme sequencing rules. This approach is highly efficient in case of agglutinative languages. However, in languages that display tendency for morpho-phonemic changes during affixation (such as Dravidian languages), this method will require an additional component of morpho-phonemic rules besides the morpheme sequencing rules.

A. ORGANIZATION OF DATA.

The analyzer based on this approach is so modeled that it analyses the inflected form of a word into suffixes and stems. It does so by making use of a root/stem dictionary (for identifying legitimate roots/stems), a list of suffixes, comprising of all possible suffixes that various categories can take (in order to identify a valid suffix), and the morpheme sequencing rules.

The Root Dictionary contains a list of roots, each with its lexical category and features. Following are samples of words from the Assamese, Bengali and Oriya root dictionaries:

1. Assamese

(a) মাহীদেউগৰাকী\NN.sg.fem ‘maternal aunt

(b) পাগলী\ADJ.fem ‘crazy’

(c) কৰ\VM ‘to do’

2. Bengali

(a) স্টার্ট\NN.0 'start'

(b) ওল্ড\ADJ.0 'old'

(c) বলা\VM 'to say'

3. Oriya

(a) ଗଛ/NN.0.0 'tree'

(b) ଗାଢ଼\ADJ.0 'bold'

(c) ଯା\VM 'go'

The Suffix List contains a list of suffixes with their morpho-syntactic feature values like gender, number, person and other relevant morphological information stored in the form of a dual field list. It deals only with inflectional suffixes not derivational. Following are samples of the Assamese, Bengali, Bodo and Oriya suffix lists.

TABLE 1: SAMPLE OF ASSAMESE SUFFIX LIST

Affix	Feature	Expansion of Abbreviations
ত	CM.Loc	Case marker, Locative
ও	Prt	Particle
টো	Cl	Classifier

TABLE 2: SAMPLE OF BENGALI SUFFIX LIST

Affix	Feature	Expansion of Abbreviations
য়	CM.loc	Case marker, Locative
টা	Prt.Def	Particle, Definite
য়েরা	Pl	Plural suffix

TABLE 3: SAMPLE OF BODO SUFFIX LIST

Affix	Feature	Expansion of Abbreviations
आव	CM.loc	Case marker, Locative
नो	Prt.emph	Particle, Emphatic
दों	Asp.prg	Aspect: Progressive

TABLE 4: SAMPLE OF ORIYA SUFFIX LIST

Affix	Feature	Expansion of Abbreviations
ରେ	CM.loc	Case marker, Locative
ଯାକ	pl	Plural suffix
ଟାଏ	Prt.def.sg	Particle- definite, singular

The Rule List provides all the possible morpheme sequences for a given category, i.e., for each category, it provides the rules identifying the ordering of suffixes.

TABLE 5: SAMPLE OF MORPHEME SEQUENCING RULES

Rules	**Expansion of abbreviations**
NN+pl+CM.ins	Noun+plural+Case marker: Instrumental
CRD+PART.emp	Cardinal+Particle: Emphatic
ORD+PART.def.sg	Ordinal+Particle: Definite, Singular
PRP+CM.gen+CM.loc	Pronoun+Case marker: genitive+ Case marker: Locative
ADJ+CM.acc	Adjective+Case marker: Accusative
VM+neg+aux.pst+sg	Verb Main +Negative+Auxiliary: Past Tense, Singular

B. THE METHOD.

Following is a Flow Chart diagram of the Morphological Analyser.

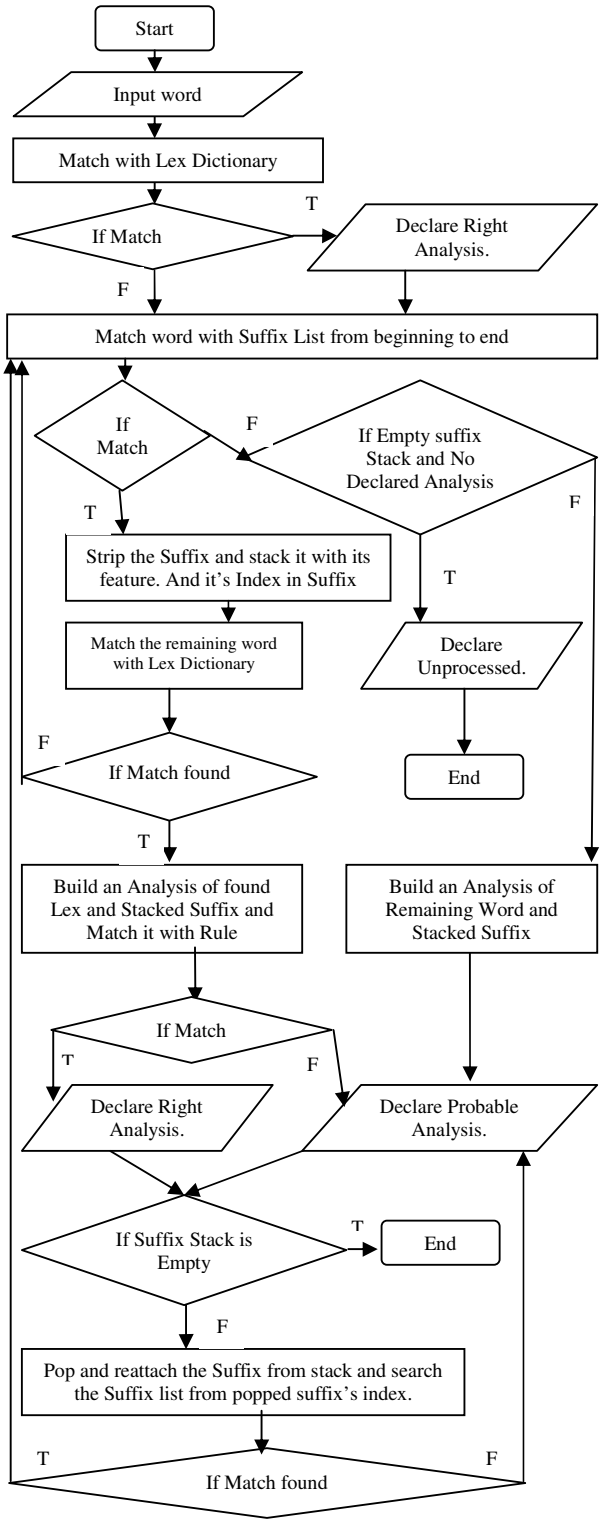


FIGURE 1: FLOW CHART DIAGRAM FOR MORPHOLOGICAL ANALYSER

The suffix stripping algorithm is a method of morphological analysis which makes use of a root/stem dictionary (for identifying legitimate roots/stems), a list of suffixes, comprising of all possible suffixes that various categories can take, and the morpheme sequencing rules. This method is economical. Once the suffixes are identified, removing the suffixes and applying proper morpheme sequencing rules can obtain the stem.

In order to identify the legitimate roots/stems, the dictionary of root/stem needs to be as exhaustive as possible. Considering this fact, the analyzer is designed to provide three types of outputs such as:

The Correct analysis: This is obtained on the basis of a complete match of suffixes, rules and the existence of the analyzed stem/root in the root dictionary.

Probable analysis: This is obtained on the basis of either a matching of the suffixes and rules, even if the root/stem is not found in the dictionary or a matching of the suffixes, but not any supporting rule or existing root in the dictionary.

Unprocessed words: These are the words which have remained unanalyzed due to either absence of the suffix in the suffix list or due to the absence of the rule in the list.

C. *Increasing the coverage (Phase 1).*

In order to increase the coverage of the system the root dictionary had to be made robust. To this end, a module has been introduced in the system, so that the roots of the probable analyses can be manually added to the root dictionary after validating them and automatically checking whether they already exist in the dictionary or not. Also, the list of unprocessed words, are manually checked and validated, after which they are added to the dictionary, with their corresponding feature values. In phase 1, this process was repeated over larger and random test corpora and with every repetition the dictionary size increased, thereby resulting in the increase in the number of correct analyses.

D. *Towards increasing the coverage (Phase 2).*

In the second phase a method has been devised to ensure that the coverage of the root/stem dictionary increases faster. Hence, the test data has been replaced by a frequency wise word list (FWL) generated from the entire available corpus of a given language. The FWL has been run on the system in blocks of 10,000 each, starting with the most frequent words to the less frequent ones in the descending order. The words which remain unanalyzed or fall under the probable analysis are first entered in the root/stem dictionary before the next block of 10,000 words are given to the system.

The logic here is simply that by first adding the most frequently occurring words in a language the overall coverage of the system shoots up manifold as compared to when entering words randomly from a corpus.

E: *Suffix and Dictionary coverage for Indian Languages.*

Details of the system coverage and the coverage of the rules and the root/stem dictionary for each of the above Languages are given below in table 6.

TABLE 6: LANGUAGE WISE COVERAGE OF THE SYSTEM

Language	Lex Dictionary Entries	Suffix-Feature pair	Rules	Coverage
Assamese	15452	216	1040	56.338 %
Bengali	12867	187	227	48.326 %
Bodo	16784	131	4379	65.82 %
Oriya	22532	127	536	70.39

CONCLUSION

The paper is about the design and development of morphological analyzers for four Indian Languages, using the suffix stripping approach. The results of the first phase of the suffix stripping approach have been fairly encouraging. It was observed, that with an average of 7000 to 8000 root entries, the affix stripping approach gives around 50% coverage. As is evident from the table 6, the coverage of the system is directly related to the size of the dictionary. We hope to expand and make the system more robust by increasing the dictionary size.

ACKNOWLEDGEMENT

We wish to thank the LDC-IL team for their support and help; but our special thanks are due to Ms. Ashmrita Gogoi,Mr. Farson Dalmari , Mr. Pramod Kumar Rout and Mr. Sankarsan Dutta for rigorously taking up the task of resource building for Assamese, Bodo, Oriya and Bengali, respectively.

REFERENCES

[1]. Bharti, A., V. Chatanya, and R. Sangal. *Natural Language Processing: A Paninian Perspective*. New Delhi: Prentice Hall. 1995.

[2]. M. L. Forcada, B. Bonev, Ortiz S. Rojas, et. al., "Documentation of the Open-Source Shallow-Transfer Machine Translation platform Apertium". 2007. Available online at: http://xixona.dlsi.ua.es/~fran/apertium2documentation.pdf.

[3]. Parameswari K. "An improvised Morphological Analyzer for Tamil: A case of implementing the open source platform Apertium". Unpublished M.Phil. Thesis. Hyderabad: University of Hyderabad. 2009.

[4]. S. Mohanty, P.K.Santi ,K.P.Das Adhikary. "Analysis and Design of Oriya Morphological Analyser: Some Tests with OriNet". *Proceeding of symposium on Indian Morphology,phonology and Language Engineering*,IIT Kharagpur. 2004.

[5]. Tyers, F. M. and Sánchez-Martínez, F. and Ortiz-Rojas, S. and Forcada, M. L. "Free/open-source resources in the Apertium platform for machine translation research and development". *The Prague Bulletin of Mathematical Linguistics. Vol.* 93. pp 67—76, 2010.

[6]. Vaidhya, Ashwini and Dipti Misra Sharma. "Using Paradigms for Certain Morphological phenomena in Marathi". *7th International Conference on NLP (ICON-2009).* New Delhi: Macmilan Publishers India Ltd., December 2009.

Sentence Boundary Disambiguation in Kannada Texts

Mona Parakh
Reader-Research Officer
ldc-monaparakh@ciil.stpmy.soft.net

Rajesha N.
Senior Technical Officer
ldc-rajesha@ciil.stpmy.soft.net

Ramya M.
Senior Technical Officer
ldc-ramya@ciil.stpmy.soft.net

Linguistic Data Consortium for Indian Languages
Central Institute of Indian Languages
Mysore, India
www.ldcil.org

***Abstract* - The proposed paper reports the work on developing a system for identifying valid sentence boundaries in Kannada texts and fragmenting the text into sentences. The task of sentence boundary identification is made challenging by the fact that the period, question marks and exclamation marks, do not always mark the sentence boundary. This paper particularly addresses the issue of disambiguating period which can be a sentence boundary marker as well as a marker of abbreviation in Kannada. This methodology is devised to fragment corpora into sentences without any intermediate tools and resources like NER or Abbreviation List.**

I. INTRODUCTION

As an important and challenging task sentence boundary disambiguation (SBD) is the problem in natural language processing of deciding where sentences begin and end. Often natural language processing tools require their input to be divided into sentences for various purposes such as building bilingual parallel corpora. "A parallel corpus is a collection of texts in two languages, one of which is the translation equivalent of the other. Although parallel corpora are very useful resources for many natural languages processing applications such as building machine translation systems, multi-lingual dictionaries and word sense disambiguation, they are not yet available for many languages of the world" [2].

In order to process information from parallel text, it is first necessary to align the two texts at some level, typically at the level of paragraph or sentence. As in Reference [1], by 'align' is meant the association of chunks of text in the one document with their translation or equivalent text in the other document. In order to align text at the level of sentences, it is important to define and identify a sentence.

For the purpose of this work, we define a Sentence as a segment of text separated by delimiters such as Exclamation mark "!", Question Mark "?", Period "." and new line character. However, these symbols do not always function as sentence delimiters; they can be used for other purposes, thereby making sentence boundary identification a non-trivial task. Sentence boundary identification is challenging because punctuation marks are often ambiguous.

Among the Indian languages Devanagari based scripts have the unique sentence boundary marker "।" known as '*poorna viraam*' (full stop) which is different from the abbreviation marker - period. Hence, in such languages segmenting sentences is a relatively trivial task. But languages like English use period as a sentence boundary maker as well as abbreviation marker. As per the English examples given in Reference [2], "a period can also be used as a decimal point in numbers, in ellipses, in abbreviations and in email-addresses. The exclamation mark in the name of a web site Yahoo! may not signify a sentence boundary and so is the question mark in Which? - the name of a magazine".

Like in English and many other languages even Kannada uses Period as a sentence boundary maker and for abbreviations. This paper attempts to handle this ambiguity of the Period in Kannada texts.

II. METHOD

Of the few papers that are available on work related to sentence boundary identification, Riley [4] uses a decision-tree based approach and claims a 99.8% performance on the Brown's Corpus. Reynar and Ratnaparkhi [3] use a maximum entropy approach to identify sentence boundaries. Some of the other common algorithms for sentence boundary identification store the Standard abbreviation as a check list; however the approach proposed in this paper assumes that since abbreviations do not form a closed set, one cannot list all possible abbreviations.

In handling the ambiguity of period in this paper, we are considering the word length as a feature. Based on the study of Kannada corpus we can safely claim that it is usually the longer words that occur at the end of sentences. If a short word occurs with a period then it is most likely either an Abbreviation or a Salutation. Based on the corpus study, a minimal threshold for word length was decided. A list was created of words having length below the threshold and which were not abbreviations. A fairly exhaustive list of some 436 such words was obtained from (approx 4.2 million words) corpus. But the list was kept open-ended in order to accommodate further additions. However, after implementing the algorithm only a few Abbreviations which were above the threshold caused over segmentation of sentences.

The detection of abbreviations is an important step in the process of sentence boundary detection. Drawing upon Reference [5] abbreviations can be categorized into three classes TRAB, ITRAB and AMAB.

a) *TRAB*: These are transitive abbreviations, i.e., abbreviations that take an object and never end the sentence. To take an example from Kannada:

Kannada script: ಮಿ. ಹರೀಶ್.

Transliteration: mi. harIsh.
Translation: Mr. Harish.

b) *ITRAB*: These are intransitive abbreviations that do not take an object. Even though Indian languages follow a relatively free word order in a sentence, normally Intransitive abbreviations do not come at the end of the sentence because, they are the subject of the sentence. Any intransitive abbreviation in the middle of a sentence will be handled by the algorithm. Following is an example from Kannada:

Kannada script: ತಮ್ಮ ಮೊಟ್ಟಮೊದಲಿನ ನಾಟಕವನ್ನು ಅ.ನ.ಕೃ. ೧೯೨೪ರಲ್ಲಿ ಬರೆದರು.

Transliteration: tamma moTTamodalina nATakavannu a.na.kx. 1924ralli baredaru.
Translation: A.Na.Kru. Wrote his first ever drama in 1924.

c) *AMAB*: These refer to abbreviations which are ambiguous, where a word is homonymous to an abbreviation.

Kannada script: ಅದನಿಲ್ಲಿ ತಾ.

Transliteration: adannilli tA.
Translation: Bring that here.

Kannada script: ತಾ. ೧೫-೦೮-೧೯೪೭

Transliteration: tA. 15-08-1947
Translation: Date. 15-08-1947

In the above example the verb 'bring' is homonymous to the standard abbreviation for 'date'. "ತಾ."/tA., could be the verb meaning "bring" occurring at the end of the sentence with a period marker or "ತಾ."/tA., could be an abbreviation for "ತಾರೀಖು"/ tArIkhu meaning "date".

III. Algorithm Design

Following is an algorithm devised to fragment the text into sentences by solving the ambiguity of period (".") as sentence marker and abbreviation in Kannada. The Algorithm uses two word lists as resource, viz. valid sentence ending word list (L1) and an ambiguous word list (L2) extracted from the corpus. This algorithm will disambiguate a period ending token as sentence ending word or abbreviation based on the token length. L1- has words having length below a threshold. L2- will have words with a length below a threshold and homonymous to an abbreviation of that language. Both L1 and L2 are extracted from corpus, and they make a small set of words. It should be noted that in this paper, the length of words refers to the length of Unicode characters and not the count of *aksharas*

Algorithm To Identify Period As Sentence Boundary

1. Preprocess the text in order to remove any space between a period (".") and its previous word.
2. Segment the text into sentences

```
1. Preprocess the text in order to remove any space between a period (".") and its  previous word.
        1.1 Open Text file
        1.2 Replace all "<space>." with "."

2. Segment the text into Sentences
 2.1 find the position of the Next Sentence Marker in the text
    2.1.1 WHILE starting position is less then Text length
    2.1.2 If the Next Immediate sentence Marker is "?" or "!" or New Line then Segment the text from Starting position to
          Sentence Marker
    2.1.3 If the Next Immediate sentence Marker is a period and not a Number before dot then
          2.1.3.1 Get the length of text between last space of text to period (Get the length of last word)
          2.1.3.2 If the Last word Length is below 5 (Threshold) then Check the word with L1
            2.1.3.2.1 If the Last word is in L1 then check the word with L2
                  2.1.3.2.1.1 If the Last word is not in L2 then Segment the text from Starting position to  Sentence Marker.
          2.1.3.3 If the Last word length is Equal or above threshold then check for the other possible dots in the Last word
            2.1.3.3.1 If there is no other possible dots in word then Segment the text from Starting position to Sentence
                      Marker.
            2.1.3.3.2 If there are other possible dots in word then check the Distance between  the end dot and the dot
                      end-but-one.
                  2.1.3.3.2.1 If Distance between the end dot and the dot end-but-one is above 5 (Threshold) then
                              Segment the text from Starting position to Sentence Marker
    2.1.4 If the Next Immediate sentence Marker is a period and a Number before dot then Segment the text from Starting
          position to Sentence Marker.
    2.1.5 End While

3. End
```

IV. Evaluation

In order to test the efficiency of the algorithm, a corpus of 7330 sentences (approx. 69000 words) was taken. Sentence Identification errors manually corrected and checked revealed that without using the algorithm and by a plain pattern matching of delimiters, a baseline accuracy of 91.33% was obtained. However, the accuracy increased to 99.14% after implementing the algorithm on the same corpus.

Out of the 7330 sentences in the corpus, the blind pattern matching without the algorithm showed errors in 636 sentences whereas after implementing the algorithm only 63 sentences were wrongly recognized. An increase of 7.81% from the baseline was noted after implementing the algorithm. The main errors occurred due to unclean corpus. Also, only a few Abbreviations which were above the threshold caused the over segmentation of certain sentences. The corpus used for the testing purpose was mainly from two domains – newspaper and literature.

V. Conclusion

In this paper we have described an algorithm for sentence boundary determination for Kannada. This methodology will hopefully be useful to resolve the problems of ambiguity of Period "." in case of text alignment tools, machine translation tools, KWIC KWOC Retrievers.

This method can be employed also for other languages. Since the check list used in the algorithm is open, it facilitates users to add more words to the list. However, depending on the language the length of the check lists may vary, as also the threshold.

Good performance has been obtained using this algorithm and it considerably increases the performance from the baseline.

Acknowledgment

Our special thanks to Prof. Kavi Narayana Murthy (CIS, UoH, Hyderabad; currently CIIL fellow) for his guidance, insightful comments and suggestions which helped us enormously in improving the evaluation work as also the paper. We are heartily thankful to our Project Head, Dr. L. Ramamoorthy, and the LDC-IL team members for their encouragement and support.

References

[1]. Harold Somers, "Bilingual parallel corpora and Language Engineering," in the Anglo-Indian Workshop on Language Engineering for South-Asian Languages, (LESAL), Mumbai, 2001.

[2]. Hla Hla Htay, G. Bharadwaja Kumar, and Kavi Narayana Murthy, "Constructing English-Myanmar Parallel Corpora," Proceedings of ICCA 2006: International Conference on Computer Applications, Yangon, Myanmar, pp 231-238, February 2006.

[3]. J. Reynar, and A. Ratnaparkhi, "A Maximum Entropy Approach to Identifying Sentence Boundaries," in Proceedings of the Fifth Conference on Applied Natural Language Processing, Washington D.C, 1997, pp. 16-19.

[4]. Riley, Michael D.. "Some applications of tree-based modeling to speech and language," in DARPA, Speech and Language Technology Workshop, Cape Cod, Massachusetts, 1989, pp. 339-352.

[5]. Trond Trosterud, Børre Gaup, Saara Huhmarniemi, "Preprocessor for Sámi language tools", The Norwegian Sami Parliament, 2004. Available online at www.divvun.no/doc/ling/preprocessor.html

CRITICAL DISCOURSE ANALYSIS: Politics and Verbal coding

Muralikrishnan.T.R
M.E.S College Marampally
Aluva, Ernakulam District
Kerala Pin: 683107
mesmurali@gmail.com

***Abstract*— Political discourse comprises all forms of communication in and by political institutions or actors and all communication with reference to political matters. Political public relations, both internal and external, news, commentary, film, talk shows, citizens' everyday talk about politics etc. are all sites of political discourse. A broad field of theoretical approaches originates in French philosophy and sociology that centre around social and political functions of discursive practices (termed Discourse Analysis). The present paper tries to discuss the close affinity shown between language and politics to work out the discursive practices apparent in public political discourses. The features of such writing/speech are taken from various political domains.**

***Keywords*— Critical Discourse Analysis (CDA), Discourse practice, Public language, representation**

I. INTRODUCTION

The objective of the paper is to illustrate how the critical discourse analysis plays a crucial role in unlocking the myths that linger in the sphere of politics and how politicians make use of their language to ensnare people for their Political discourse comprises all forms of communication in and by political institutions or actors and all communication with reference to political matters. Political public relations, both internal and external, news, commentary, film, talk shows, citizens' everyday talk about politics etc. are all sites of political discourse. The shift from 'Fordist' economy to 'flexible accumulation' (both technological innovation in the diversification of production and flexibility of labour), transnational movement of production units, opening up of new experiences owing to information technology and media, cultural transformation due to circulating signs liberated from fixed boundaries as represented in postmodernist theory, are in total a phase of late modernity. This phenomena encapsulates good, bad and the ugly i.e. this creates new possibilities and opportunities for many at the same time this can also cause considerable disruption and suffering. But the entire experience is communicated to be perceived as something inevitable. The relevance of CDA (Critical Discourse Analysis) is that it can expose the transformations in language and discourse favouring such trends. It can enlighten how the discourse shapes and reshapes the given reality. CDA has set out a dialectical view of the relationship between discourse and other facets of the social world.

II. A THEORETICAL FRAMEWORK

Critical Discourse Analysis or CDA is an approach to discourse analysis in which two senses of the term discourse- the linguistic sense and the critical theorist's sense-are equally relevant. The 'critical' in CDA refers to a way of understanding the social world drawn from critical theory. Within that paradigm reality is understood as *constructed*, shaped by various social forces. These, however are frequently naturalized- in every day discourse, as opposed to critical discussions of it, reality is presented not as the outcome of social practices that might be questioned or challenged, but as simply "the way things are". Naturalization obscures the fact that 'the way things are' is not inevitable or unchangeable. It both results from particular actions and

serves particular interests. According to van Dijk, CDA "is a type of discourse analytical research that primarily studies the way social power abuse, dominance, and inequality are enacted, reproduced, and resisted by text and talk in the social and political context. With such dissident research, critical discourse analysts take explicit position, and thus want to understand, expose, and ultimately to resist social inequality"[1]. The central claim of CDA is that the way certain realities get talked or written about- that is, the choices speakers and writers make in doing it-are not just random but ideologically patterned. Norman Fairclough uses discourse analysis techniques to provide a political critique of the social context- from a Marxist viewpoint. He defines what he calls critical language study thus: "Critical is used in the special sense of aiming to show up connections which may be hidden from people- such as the connections between language, power and ideology…critical language study analyses social interactions in a way which focuses upon their linguistic elements and which sets out to show up their generally hidden determinants in the system of social relationships, as well as hidden effects they may have upon that system" [2]. He is candid about his own starting point and about his own political purpose: "I write as a socialist with a genuinely low opinion of the social relationships in my society and a commitment to the emancipation of the people who are oppressed by them. This does not, I hope, mean that I am writing political propaganda. The scientific investigation of social matters is perfectly compatible with committed and 'opinionated' investigators (there are no others!) and being committed does not excuse you from arguing rationally or producing evidence for your statements." (Ibid: 5) One sees "discourse" as an abstract noun denoting language in use as a social practice with particular emphasis on larger units such as paragraphs, utterances, whole texts or genres. The other identifiable meaning is "Discourse" as a countable noun denoting 'a practice not just of representing the world, but of signifying the world, constituting and constructing the world in meaning'[3]. For some, discourse analysis is a very narrow enterprise that concentrates on a single utterance, or at most a conversation between two people. Others see discourse as synonymous with the entire social system, in which discourses literally constitute the social and political world. As the concept of discourse has been employed in the social sciences, it has acquired greater technical and theoretical sophistication, while accruing additional meanings and connotations. Positivists and empiricists argue that discourses are best viewed as 'frames' or 'cognitive schemata' by which they mean 'the conscious strategic efforts by groups of people to fashion shared understandings of the world and of themselves that legitimate and motivate collective action'[4]. By contrast, realist accounts of discourse place much greater emphasis on what they call the ontological dimensions of discourse theory and analysis. Discourses are regarded as particular objects with their own properties and powers, in which case it is necessary for realists 'to focus on language as a structured system in its own right', and the task of discourse analysis is to unravel 'the conceptual elisions and confusions by which language enjoys its power' [5]. Marxists stress the way in which discourses have to be explained by reference to the contradictory processes of economic production and reproduction. In this perspective, discourses are normally viewed as ideological systems of meaning that obfuscate and naturalize uneven distributions of power and resources. This means that discourse analysis has the critical task of exposing the mechanisms by which this deception operates and of proposing emancipatory alternatives (Althusser 1971; Pêcheux 1982). Giddens's (1984) theory of society differs from positivist, realist and Marxist accounts in that he stresses the centrality of human meaning and understanding in explaining the social world. His explicitly 'hermeneutically informed social theory' thus places greater emphasis on the actions and reflexivity of human agents in reproducing and changing social relationships. Fairclough takes up this theme of 'the duality of social structure and human agency' by insisting that there is a mutually constituting relationship between discourses and the social systems in which they

function. The task of discourse analysis is thus to examine this dialectical relationship and to expose the way in which language and meaning are used by the powerful to deceive and oppress the dominated. Finally, post-structuralists and post-Marxists such as Jacques Derrida, Michel Foucault, Ernesto Laclau and Chantal Mouffe put forward much more comprehensive concepts of discourse. They go further than the hermeneutical emphasis on social meaning by regarding social structures as inherently ambiguous, incomplete and contingent systems of meaning. Derrida (1982) argues for a conception of discourse as text or writing, in which all human and social experience is structured according to the logic of *différance*; while Foucault (1981, 1991) intends to show the connection between 'discursive practices' and wider sets of 'non-discursive' activities and institution. Laclau and Mouffe (1985, 1987) deconstruct the Marxist theory of ideology and draw upon post-structuralist philosophy to develop a concept of discourse that includes all practices and meanings shaping a community of social actors. In these perspectives, discourses constitute symbolic systems and social orders, and the task of discourse analysis is to examine their historical and political construction and functioning.

Some of Foucault's ideas have been very influential on the different approaches to critical discourse analysis. The central task for Michel Foucault was to write a history of expounding problems as a critique and destruction of western thinking, which had always focused on what it means to be human being instead of how it is to be a human being. Although human beings are acting in their lives, they are not the subject of these actions, but products of discursive practices. Objects are not social facts, but how subjects bring things to presence through language (objectification). Therefore a relation between power and language can be stated and subjects must be seen as social constructions, produced through social discourses which position them in a field of power relations. While critical thinking focuses on our ability to gain access to language (through knowledge), Foucault focuses on how technologies of calculation produce calculable and empowered subjects because of their inscription into force by technology. Therefore his attack of the central importance of the subject can be seen as his major interest. Foucault defines discourses as knowledge systems that inform social and governmental technologies. These technologies constitute power in society. Power does not come from outside, but is in us, the dead subjects, who are ruled by our own creations and constructions: the technologies and techniques of power in social institutions. Thus Michel Foucault opposes the concept of ideology because it is implicated in unacceptable universal truth claims and rests on a humanist understanding of the individual subjects [6]. Foucault saw power in contrast to Marxist theorists to whom power was an instrument of class dominance originated from economic interests, as something incorporated in numerous practices and technologies and not attached to certain agents or interests. In Foucault's concept of power the word "how" is the basic key word of analysis. Discourses are expressions of power relations and refer to all that can be thought, written or said about a particular topic or thing. They draw on historically prior texts and are generated by combination of and with other discourse and texts (interdiscursivity and intertextuality). Discourse analysis is concerned with the rules (practices, technologies), which make others not at particular times, places and institutional locations. Certain rules of formations of objects, subjects, concepts and strategies are constituted in a given discursive formation and can be seen as the basic rules of discursive order. Objects and subjects of discourses are organized in relation to particular concepts, which involve social and political experiences about the modalities relating subjects and objects to each other. These modalities of relating objects, subjects and concepts, Foucault label strategies. With strategies he does not mean particular intentions of goals of the subjects, but topical choices, which interrelate subjects, objects and concepts in discourses to each other and across different discourses. The analysis of rules of formations of objects, subjects, concepts and topical choices can be seen as a fundamental approach to discourse analysis. It illuminates

which objects, subjects, concepts or topics are banned from a particular discourse and how the relations between those elements are established in this discourse. The link between practice and speaking lies in his concept of 'power/knowledge'. In the modern age, a great deal of power and social control is exercised not by brute physical force or even by economic coercion, but by the activities of 'experts' who are licensed to define, describe and classify things and people. As Deborah Cameron says, "Words can be powerful: the institutional authority to categorize people is frequently inseparable from the authority to do things to them" [7]

Fairclough and Wodak (1997) analyze an extract from a radio interview with former British Prime Minister, Margaret Thatcher, with reference to 'eight principles of theory or method' [8], which are:

1. CDA addresses social problems
2. Power relations are discursive
3. Discourse constitutes society and culture
4. Discourse does ideological work
5. Discourse is historical
6. The link between text and society is mediated
7. Discourse analysis is interpretative and explanatory
8. Discourse is a form of social action.

According to Wodak and Ludwig (1999) viewing language this way entails three things at least. First, discourse "always involves power and ideologies. No interaction exists where power relation do no prevail and where values and norms do not have a relevant role" [9]. Second, "discourse…is always historical, that is, it is connected synchronically and diachronically with other communicative events which are happening at the same time or which have happened before" [*ibid*]. The third feature of Wodak's approach is that of interpretation. According to Wodak and Ludwig, "readers and listeners, depending on their background knowledge and information and their position, might have different interpretations of the same communicative event" [*ibid*]. Therefore, Wodak and Ludwig assert that "THE RIGHT interpretation does not exist; a hermeneutic approach is necessary. Interpretations can be more or less plausible or adequate, but they cannot be true"[emphasis in original] (*ibid*). Chouliaraki and Fairclough [10] posit that CDA has a particular contribution to make. According to them, the recent economic and social changes "are to a significant degree…transformations in the language, and discourse", thus CDA can help by theorizing transformations and creating an awareness "of what is, how it has come to be, and what it might become, on the basis of which people may be able to make and remake their lives". In this approach of CDA, there are three analytical focuses in analyzing any communicative event (interaction). They are *text* (e.g. a news report), *discourse practice* (e.g. the process of production and consumption) and *sociocultural practice* (e.g. social and cultural structures, which give rise to the communicative event [11].

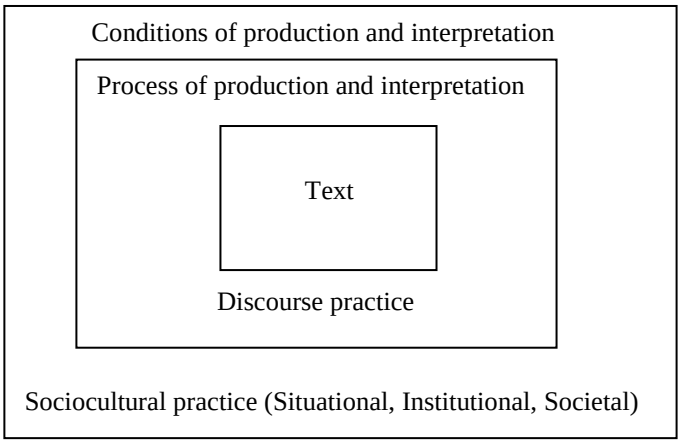


FIG. 1: COMMUNICATIVE EVENT

Fairclough has proposed a framework for analyzing a communicative event. The first analytical focus is on text. Analysis of text involves linguistic analysis in terms of vocabulary, grammar, semantics, the sound system and cohesion- organization above the sentence level. Following SFL, Fairclough also views text from a multifunctional perspective. According to him, any sentence in a text is analyzable in terms of the articulation of these functions,

which he has relabeled- *representations, relations and identities*:

- Particular representations and contextualisations of social practice (ideational function)- perhaps carrying particular ideologies.
- A particular construction of the relationship between writer and reader (as, for instance, formal or informal, close or distant)
- Particular constructions of writer and reader identities (for example, in terms of what is highlighted-whether status and role aspects of identity, or individual and personality aspects of identity) [*ibid*]

Linguists preferred to use the 'discourse' to refer to language in use. In studying discourse they focus on written text, on spoken utterance, and on the processes whereby individuals process texts and utterances. On the other hand, social scientists in the 1970 and 80s were influenced by the way the term 'discourse' is used in European literary and social criticism by writers such as the French philosopher Michel Foucault. Some linguists, as Fairclough, concerned with the critical analysis of language use in relation to politics have adopted these ideas.

Terry Locke [12] has summed up the ways in which CDA can be understood. According to him CDA:

- views a prevailing social order as historically situated and therefore relative, socially constructed and changeable.
- views a prevailing social order and social processes as constituted and sustained less by the will of individuals than by the pervasiveness of particular constructions or versions of reality-often referred to as discourses.
- views discourse as coloured by and productive of ideology (however 'ideology' is conceptualized)
- views power in society not so much as imposed on individual subjects as an inevitable *effect* of a way particular discursive configurations or arrangements privilege the status and positions of some people over others.
- views human subjectivity as at least in part constructed or inscribed by discourse, and discourse as manifested in the various ways people are and enact the sorts of people they are.
- views reality as textually and intertextually mediated via verbal and non-verbal language systems and texts as sites for both the inculcation and the contestation of discourses.
- views the systematic analysis and interpretation of texts as potentially revelatory of ways in which discourses consolidate power and colonize human subjects through often covert position calls.

III. LANGUAGE AND POLITICS

The present paper tries to discuss the close affinity shown between language and politics to work out the discursive practices apparent in public political discourses. Politics is concerned with power- the power to make decisions, to control resources, to control other people's behaviour and often to control their values. The acquisition of power, and the enforcement of ones belief systems can be achieved in a number of ways; one obvious method is through physical coercion or by indirect means of coercion through the legal system. However it is often much more effective to persuade people to act voluntarily in the way one wants, i.e., to exercise power through the manufacture of consent or at least acquiescence towards it. To achieve this an ideology needs to be established. One which make the beliefs which one wants people to hold appear to be "common sense", thus making it difficult for them to question that dominant ideology. It was Louis Althusser who wondered how the vast majority of people had been persuaded to act against their own best interests, since they worked long hours at laborious task and lived in poverty while a very small number of people made enormous amounts of money from their labour and enjoyed lives of luxury. In order to explain why the impoverished

majority didn't just refuse to work in this system and overthrow the rich minority, Althusser reasoned that the poor had been persuaded that this state of affairs was 'natural' and nothing could be done to change it.

In the discussion that follows we consider the language of 'career' politicians who play a significant part in the political scenario of the state. The first comment that is to be made about political discourse is that it is not simply a genre, but a class of genres defined by a social domain, namely that of politics. However it is difficult to create a separation from other domains and in this case the boundaries are fuzzy. To make it simple, as a beginning, political discourse is the discourse of politicians. The range in this study has to be limited to the 'professional' realm of the activities of 'career' politicians. The activity considered must be in the public sphere and the discourse must be produced by the speaker in his / her professional role of a politician and it should be based in an institutional setting. Thus the discourse is political when it accomplishes a political act in a political institution, such as governing, legislation, electoral campaigning and so on.

Because political discourse is a broad category, studies on political language have included investigation into very different sub-genres such as electoral language, party political language, the language of diplomacy and international relations, the language of social conflict, the language of parliament and so on. Language is a means of communication, a means of presenting and shaping argument and political argument is ideological, in that it comes from a series of beliefs. Language is not something somehow separate from the ideas it contains, but the way language is used says a great deal about how the ideas have been shaped. When analyzing the language of a political text, therefore, it is important to look at the way the language reflects the ideological position of those who have created it, and how the ideological position of the readers will affect their response too.

Chilton. P. identifies two approaches viz., descriptive and critical, for dealing with this issue of politics and language. Descriptive approach relies on re describing rhetorical devices, the verbal behaviour of politicians and their ideology; whereas critical approaches incorporate social theories dealing with the relationship between language and power, control and dominance and orders of discourse. A detailed discussion on the above will be taken in due course. Chilton. P, while elaborating on the modern descriptive approaches, have recast the traditional rhetorical aspects like persuasive, deceptive and manipulative, in terms of phonological, syntactic, lexical, semantic, pragmatic and textual levels of description. Thus on the phonological level can be placed devices of alliteration, assonance and rhythm; on the syntactic level, the use of agent-less passive; on the lexical level, emphasis will be on 'jargon' words- that is words characteristic of some closed group of speakers, neologisms, acronyms and word formation; on the semantic level the interest is in semantic reconstruction and shifts arising from, for e.g., paraphrasing, and euphemism. On the textual and pragmatic levels, commentators have noted modes of argumentation. The descriptive strand of study tends to take an epistemological position that is close to positivism. It tends to treat the political language phenomena it is submitting to scrutiny as neutral independent facts. Whereas in a critical approach, it assumes a different conception of politics- a conception which emphasizes the importance of power from the point of view of the subject citizen and assumes connections between the macro structures of state institutions and the micro structures of everyday person to person relation ships and interactions. Jurgen Habermas and Michel Foucault have been influential in the evolution of this thought. Habermas is associated with Frankfurt School 'Critical theory' and views the analysis of social practices, including linguistic ones as a rational enterprise whose purpose is emancipation. 'Distorted Communication' derives in the Habermasian view, from unequal access to the communication process, which itself is a function of the exercise of power. Linguists preferred to use the 'discourse' to refer to language in use. In studying discourse they focus on written text, on spoken utterance, and on the processes whereby individuals process texts and utterances. On the other

hand, social scientists in the 1970 and 80s were influenced by the way the term 'discourse' is used in European literary and social criticism by writers such as the French philosopher Michel Foucault. Some linguists, as Fairclough (1989), concerned with the critical analysis of language use in relation to politics have adopted these ideas. They emphasize prominently notions like;

1. The relationship between language and power: control and dominance, it is claimed, are exercised increasingly in the modern period by linguistic means.
2. The pervasiveness of power: control and dominance are everyday phenomena found in encounters of many kinds.
3. The relationship between linguistic and non-linguistic practices.
4. Orders of discourse: types of talking and writing play different parts in different institutions of a society.

Van Dijk (Ref 1) has noted that political discourse is not a genre, but a class of genres defined by a social domain, namely that of politics. Thus, government deliberations, parliamentary debates, party programmes and speeches by politicians are among the many genres that belong to the domain of politics. The discourse must be produced by the speaker in her/his professional role of a politician and in an institutional setting. While analyzing their topic and style, topics are usually about events in the public sphere and style incorporates many rhetorical features (metaphors, euphemism, hyperbole etc). It also allows inferences about the cognitive, social and especially political functions of such discourse.

The concept of ideology is crucial in political science and since ideologies are defined in terms of basic beliefs shared by the members of groups, this also means that political discourse is the site where politicians' multiple ideological identities are enacted. Political and ideological discourse analysis is usually based on individual discourses, so it will not be strange at all to find influence of various ideologies. The social identity of politicians will also be defined by such categories as membership devices, activities, aims and goals, norms and values, relations to other groups and resources or 'capital'. Van Dijk (Ref 1) has roughly defined the ideological self-identity of politicians as professionals.

a. Identity criterion: Election to political office.
b. Activities: 'Doing' politics (represent citizens, legislate etc.)
c. Aim: Govern country, state or city etc.
d. Norms, values: Democratic values, honesty etc.
e. Position, relation to other groups: relation with constituents etc.
f. Resource: Political power.

Thus, if politicians regularly criticize other politicians for 'not listening to the voice of the people', as is often the case in populist political discourse, then we may assume that the basic activities and norms defining the ideology of politicians involve 'listening to the voice of the people'. If there are political ideologies, then they must specifically apply in the domain of politics, and organize political attitudes and political practices. If we focus on politicians, we shall usually have at least two ideologies as expressed in their text and talk: viz., firstly professional ideologies that underlie their functioning as politicians and secondly the socio-political ideologies they adhere to as members of political parties or social groups. Thus ideology, politics and discourse form a triangle that poses interesting theoretical and analytical questions. Defined as socially shared representations of groups, ideologies are the foundations of group attitudes and other beliefs, and thus also control the biased personal mental models that underlie the production of ideological discourse.

The point of ideological discourse analysis is not merely to discover underlying ideologies, but to systematically link structures of discourse with structures of ideologies. If ideologies are acquired, expressed, enacted and reproduced by discourse, this must happen through a number of discursive structures and strategies. In theory and depending on context, any variable structure of discourse may be ideologically 'marked'. It should be stressed that ideologies may only influence the contextually variable structures of discourse.

Obviously the obligatory, grammatical structure cannot be ideologically marked because they are the same for all speakers of the language and in that sense ideologically neutral. However, there may be some debate on whether some general grammatical rules are really ideologically innocent. Some variable structures are more ideologically 'sensitive' than others. Syntactic structures and rhetorical figures such as metaphors, hyperboles or euphemisms are used to emphasize or de-emphasize ideological meanings, but as formal structures they have no ideological meaning. A general tendency among the organization of ideological discourses is the strategy of positive self-presentation (boasting) and negative other-presentation (derogation). There are many discursive ways to enhance or mitigate our / their good / bad things, and hence to mark discourse ideologically.

The concept of "public language" [13] is significant in understanding political discourse. Public language validates established beliefs and strengthens the authority structures of the polity or organization in which it is used. It is therefore the language form supporter of regimes or organizations rely on to demonstrate to others and to themselves that they deserve support to minimize guilt, to evoke feelings in support of the guilt, to evoke feelings in support of the polity, and to engender suspicion of alternatives and of people identified as hostile. It can take many political forms. As Edelman says, "Exhortation to patriotism and to support for the leader and his/her regime" are obvious ones. Less obvious forms, according to him are;

1. Terms classifying people according to the level of their merit, competence, or authority.
2. Terms that implicitly define an in-group whose interests conflict with those of other groups.
3. Presentational forms that justify actions and policies. [14]

Jason Jones and Jean Peccei [15] have outlined some strategies employed by politicians to influence people's political and ideological views for their own advantage.

- Presupposition (background assumptions embedded within a sentence or phrase)
- Implicature (dependent more on shared knowledge between the speaker and hearer)
- Persuasive language (making use of metaphor, euphemism, three-part-statement, parallelism, pronouns for identification)

Marlene Caroselli [16] studied the language outstanding leaders use when they address their audiences and has identified the following elements.

1. Display a positive attitude toward the communication process.
2. Know how to tell a good story
3. They admit to human failures
4. Display emotion
5. Improve the status quo
6. Use challenging statements to inspire, motivate and direct energy toward the best possible outcomes
7. Use personal stories and anecdotes
8. Are forthright to declare what they stand for
9. Use parallelisms in sentence structure,
10. Use the appropriate style.

Heatherington [17] lists the sorts of language exploitation indulged by politicians.

a. Good feelings- evocating feelings of patriotism (vote for Us is patriotic, good, while a vote for Them is non-patriotic, treacherous); direct flattering of audience ("the sensible voter"); reference to "the record" (his voting record, wisdom, skill and their benefits for his audience).

b. Bad feelings- evocating emotions of fear, anger and scorn (against the values mooted by the opposition)

c. Fog- use of buzz or fad words with a high fog index, that is, abstract, non-referential and often polysemous signs. This technique appears most often when a politician is in trouble and trying to justify his behaviour "to the folks back home"; the fog makes it nearly impossible to assign responsibility to anyone, least of all to the speaker.

Heatherington (*ibid*) also identifies three characteristics that often distinguish propaganda from ordinary persuasion.

1. A consistent choice of loaded language over more value-free language.
2. A heavy use of stock phrases.
3. A flavour of having the answers ready made.

IV. ANALYSIS OF ENCODING TECHNIQUES: DISCURSIVE STRATEGIES AND PRACTICES

In this context, the word 'practice' requires some elaboration. All practices involve configurations of diverse elements of life and therefore diverse mechanisms. A practice can be understood both as a social action, what is done in a particular time and place, and as what has hardened into relative permanency- a practice in the sense of a habitual way of living. Chouliariki and Fairclough (Ref.10) have identified three main characteristics: "First, they are forms of production of social life, not only economic production but also in for instance, the cultural and political domain. Second, each practice is located within a network of relationship to other practices, and these 'external' relationships determine its 'internal' constitution. Third, practices always have a reflexive dimension: people always generate representations of what they do as part of what they do". Here one has to consider the factor of power in the sense of domination at the level of particular practice, where, as Giddens (1984) and Bourdieu (1991) have pointed out, subjects are positioned in relation to others such that some are able to incorporate the agency of others into their own actions and so reduce the autonomous agentive capacity of the latter. Gramsci's concept of hegemony is helpful in analyzing relations of power as domination. Hegemony is relations of domination based upon consent rather than force (coercion), involving the naturalization of practices and their social relations as well as relations between practices. Ideologies are constructions of practices from particular perspectives which help to level out the inner contradictions and antagonisms of practices in ways which accord with the interest and projects of domination. A discourse is way of signifying a particular domain of social practice from a particular perspective. One can say that the discourse of one practice colonises that of another, or that the latter appropriates the former, depending on how power relations are expressed as relations between practices and discourses. So ideologies are domination-related constructions of a practice which are determined by specifically discursive relations between that practice and other practices. The figure given in Wodak and Meyer [18] gives the 'Fields of action' in the political area. This comprises of legislation, self-presentation, the manufacturing of public opinion, developing internal party consent, ad campaign, vote getting, governance and execution, control and expression of dissent.

Law making political procedure	Formation of public opinion and self presentation
laws	press releases
bills	conferences
amendments	interviews
speeches & contributions of MPs	talk shows
regulations	lectures and
recommendations	articles, books
prescriptions	commemorative speeches
guidelines	inaugural addresses
etc	etc

Fig: 2 *Selected dimensions of discourse as social practice*

FIELDS OF ACTION	
Party-internal development of an informed opinion	Political advertising, marketing and propaganda
party programmes, declarations, statements	election
speeches of principles	slogans
speeches on party	speeches in
conventions	campaigns
etc	announcements
	posters
	election brochure
	direct mailing
	fliers
	etc

Fig: 3 *Selected dimensions of discourse as social practice*

	FIELD OF CONTROL
Political executive and administration	Political executive and administration
decisions	decisions
inaugural speeches	inaugural speeches
coalition papers	coalition papers
speches of ministers	speches of ministers
heads	heads
governmental answers	governmental answers
p.q.	p.q.

Fig: 4 *Selected dimensions of discourse as social practice*

There are several discursive elements and strategies which deserve to receive special attention. They include questions such as (Ref.18);

1. How are persons named and referred to linguistically?
2. What traits, characteristics, qualities and features are attributed to them?
3. By means of what arguments and argumentation schemes do specific persons or social groups try to justify and legitimize the exclusion, discrimination, suppression and exploitation of others?
4. From what perspective or point of view are these labels, attributions and arguments expressed?
5. Are the respective utterances articulated overtly? Are they intensified or are they mitigated?

These help in identifying the positive *self* representation and negative *other* representation.

V. CONCLUSION

CDA is mainly interested in the role of discourse in the instantiation and reproduction of power and power abuse, and hence particularly interested in the detailed study of the interface between structures of discourse and the structures of power. Issues of politics and society are thus not merely abstract systems of social inequality and dominance, but they actually 'come' down in the forms of everyday life, namely through the beliefs, actions and discourses of group members. CDA is specifically interested in the discursive dimensions of these abuses, and therefore must spell out the detailed conditions of the discursive violations of human rights, when newspapers publish biased accounts about the marginalized, when managers engage in or tolerate sexism in the company or organization, or when legislators enact neo-liberal policies that make the rich richer and the poor poorer.

REFERENCES

[1] Van Dijk. T. 2001. 'Ideology and discourse. A multidisciplinary introduction'. Internet course for the Oberta de Catalunya (UOC). Retrieved Nov 26, 2004 from http://www.discourse-in-society.org/ideo-dis2.htm

[2] Fairclough, N. *Language and Power*. London: Longman. 1989.

[3] Fairclough, N. *Discourse and Social Change*. Cambridge: Polity. 1992.

[4] McAdam, D. McCarthy, J.D. & Zald, M.N. (eds) Comparative Perspectives on Social Movements: Political Opportunities, Mobilizing Structures, and Cultural Framings. Cambridge: Cambridge University Press. 1996

[5] Parker, Ian. Discourse Dynamics: critical analysis for social and individual psychology. London: Routledge. 1992.

[6] Foucault, Michel. *Power/Knowledge*: *Selected Interviews and Other Writings, 1972-77*. New York: Pantheon Books. 1980.

[7] Cameron, D. *Working with spoken discourse*. London: Sage. 2001.

[8] Fairclough Norman & Ruth Wodak 'Critical Discourse Analysis', In Teun A. Van Dijk (ed.) *Discourse as Social Interaction* Vol 1: 258-284. London: Sage .1997.

[9] Wodak, R. and Ludwig, Ch. (eds.) Challenges in a Changing World: Issues in Critical Discourse Analysis. Vienna: Passagenverlag. 1999.

[10] Chouliaraki, L. and Fairclough,N. *Discourse in Late Modernity: rethinking critical discourse analysis*. Edinburgh: Edinburgh University Press. 1999.

[11] Fairclough, Norman .*Critical Discourse Analysis: The critical study of language*. London and New York: Longman. 1995.

[12] Locke, Terry. *Critical Discourse Analysis*. London: Continuum. 2004.

[13] Bernstein, B. *Class, Codes and Control, Volume 2*. London: Routledge. 1975.

[14] Edelman, Murray. Political Language: Words that succeed and Policies that fail. New York: Academic Press. 1977.

[15] Jones, Jason and Jean Stilwell Pecci. 'Language and Politics'. In Linda Thomas, Joanna Thornborrow et al (eds.) *Language, Society and Power*: *An Introduction*. London: Routledge. 2003.

[16] Caroselli, Marlene. *Leadership Skills for Managers*. New York: McGraw Hill. 2000.

[17] Heatherington, Madelon.E. *How Language Works*. Massachusetts: Winthrop. 1980.

[18] Wodak, R. 'What CDA is about- a summary of its history, important concepts and its developments'. In Wodak, R. & Meyer, M. (eds.) *Methods of Critical Discourse Analysis*. London: Sage 2001.

A First Step Towards Parsing of Assamese Text

Navanath Saharia
Department of CSE
Tezpur University
Assam, India 784028
nava.nath@yahoo.in

Utpal Sharma
Department of CSE
Tezpur University
Assam, India 784028
utpal@tezu.ernet.in

Jugal Kalita
Department of CS
University of Colorado
Colorado Springs, USA 80918
kalita@eas.uccs.edu

***Abstract*—Assamese is a relatively free word order, morphologically rich and agglutinative language and has a strong case marking system stronger than other Indic languages such as Hindi and Bengali. Parsing a free word order language is still an open problem, though many different approaches have been proposed for this. This paper presents an introduction to the practical analysis of Assamese sentences from a computational perspective rather than from linguistics perspective. We discuss some salient features of Assamese syntax and the issues that simple syntactic frameworks cannot tackle.**

***Keywords*-Assamese, Indic, Parsing, Free word order.**

I. Introduction

Like some other Indo-Iranian languages (a branch of Indo-European language group) such as Hindi, Bengali (from Indic group), Tamil (from Dardic group), Assamese is a morphologically rich, free word order language. Apart from possessing all characteristics of a free word order language, Assamese has some additional characteristics which make parsing a more difficult job. For example one or more than one suffixes are added with all relational constituents. Research on parsing model for Assamese language is purely a new field. Our literature survey reveals that there is no annotated work on Assamese till now.

In the next section we will present a brief overview of different parsing techniques. In section III we discuss related works. Section IV contains a brief relevant linguistic background of Assamese language. In section V we discuss our approach we want to report in this paper. Section VI conclude this paper.

II. Overview of parsing

The study of natural language grammar dates back at least to 400 BC, when Panini described Sanskrit grammar, but the formal computational study of grammar can be said to start in the 1950s with work on context free grammar(CFG). Parsing is a problem in many natural languages processing tasks such as machine translation, information extraction, question answering etc. It is the process of automatically building syntactic analysis of a sentence in terms of a given grammar and lexicon; and syntax is the name given to the study of the form, positioning, and grouping of the elements that go to make up sentences. The result may be used as input to a process of semantic interpretation. The output of parsing is something logically equivalent to a tree, displaying dominance and precedence relation between constituents of a sentence. Now-a-days there are several dimensions to characterize the behaviour of parsing technique, for example- depending on search strategy (such as Top-down, bottom-up parsing), statistical model used (such as Maximum Entropy model), Grammar formalism used (such as Paninian framework) etc. Among them most successful linguistically motivated formalisms are- Combinatory Categorial Grammar (CCG), Dependency Grammar(DG)[1], Lexical Functional Grammar (LFG) [2], Tree-Adjoining Grammar (TAG) [3], Head-Driven Phrase Structure Grammar (HPSG) [4], Paninian Grammar (PG) [5] and Maximum Entropy model (EM) [6].

The Department of Computer science & Engineering, Tezpur University is funded by university grant commission (UGC)'s departmental research support (DRS) Phase-I under special assistance programme (SAP).

III. Existing work

Reference [7], reported (Table I) word order variability that some language allow.

TABLE I
WORD ORDER VARIATION TABLE.

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

Our literature survey reveals that a majority of the parsing techniques are developed solely for the English language and might not work for other languages.Much work has been done in different languages in different aspect of parsing, but most of these approaches can not be applied to Indian language context. The main reason is most of the Indian languages are highly inflectional, relatively free word order and agglutinative. Unlike fixed word order language such as English, in morphologically rich free word order languages the preferable linguistics rule set is too large, which may not be handled using the approaches like PSG, LFG[2] etc. Among the reported formalisms, only CCG, PG and DG have literal evidence to apply on free word order languages.

An approach for Indian language parsing is Paninian framework which was developed in IIT, Kanpur. First it was designed only for free word order languages basically Hindi, afterward it was extended to other free word order language

such as Bangla, Tamil etc., but no attempt was made to build a parser for Assamese.

Among the more recent works [8], [9], [10] has focus on dependency parsing. Dependency grammar is an asymmetrical relation between a head and a dependent. Dependency grammar is a set of rules that describes the dependencies. Every word (dependent) depends on another word (head), except one word which is the root of the sentence.Thus a dependency structure is a collection of dependencies for a sentence and dependency parsing depends critically on predicting head-modifier relationship.

A classifier based dependency parser was proposed by Sagae and Lavie [11], that produces a constituent tree in linear time. The parser uses a basic bottom-up shift-reduce stack based parsing algorithm like Nivre and Scholz[12] but employs a classifier to determine parser actions instead of a grammar. Like other deterministic parsers (unlike other statistical parser), this parser considers the problem of syntactic analysis separately from part-of-speech (POS) tagging. Because the parser greedily builds trees bottom-up in a single pass, considering only one path at any point in the analysis, the task of assigning POS tags to word is done before other syntactic analysis. This classifier based dependency parser shares similarities with the dependency parser of Yamada and Matsumoto [13] that it uses a classifier to guide the parsing process in deterministic fashion, while Yamada and Matsumoto uses a quadratic run time algorithm with multiple passes over the input string.

A language-wise survey (Table II) shows that Nivre's parser was implemented in a variety of languages, like relatively free word order language (Turkish), inflectionally rich language (Hindi), fixed word order language (English), and relatively case-less and less inflectional language (Swedish), whereas Paninian grammar framework was implemented only for Indian language context and CCG approach was implemented for Dutch, Turkish and English Language. Other mostly implemented parsers are Collin's and Mc-Donald's parser.

TABLE II
LANGUAGE-WISE SURVEY OF IMPLEMENTED PARSER.

Nivre's Parser	English[12] Czech[14] Swedish[15] Chinese[16] Bulgarian[17] Turkish[18] Hindi[8]
Collin's Parser	English[19] Czech[20] Spanish[21] Chinese[22] German[23]
Mc Donald's Parser	English[24] Czech[24] Danish[25]
CCG Framework	English[26] Dutch[27] Turkish[28]

IV. ASSAMESE AS A FREE WORD ORDER LANGUAGE

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

Typical Assamese sentences can be divided into two parts: Subject(S) and Predicate(P). Predicate may again be divided into following constituents- object(O), verb(V), extension(Ext) and connectives(Cv). A minimum sentence may consist of any one of S, O, V, Ex or even in a connected discourse. Table III shows some single constituent sentences of Assamese. Table IV shows, some two-constituent sentences that may also occur in any order.

TABLE III
SINGLE CONSTITUENT SENTENCES. (TF: TRANSLITERATED ASSAMESE FORM, ET: APPROXIMATE ENGLISH TRANSLATION)

N —	নমস্কাৰ।	TF: *namoskAr.*	
PN—	মই।	TF: *mai*	ET: I.
V —	আহা।	TF: *ahA*	ET: come.
PP—	আৰু।	TF: *aAru*	ET: and.

TABLE IV
TWO CONSTITUENT SENTENCES.

PN+V	মই আহিছো। TF: *mai aAhiso* EF: I have come.	V+PN	আহিছো মই। TF: *aAhiso mai*
N+V	কিতাপখন পঢ়িলো। TF: *kitApkhan parhilo* EF: (I) have read the book.	V+N	পঢ়িলো কিতাপখন। TF: *parhilo kitApkhan*
Adj+V	ভাল গাইছে। TF: *vAl gAICe* EF: Sang well.	V+Adj	গাইছে ভাল। TF: *gAICe vAl*
PP+V	যদি আহা ! TF: *yadi aAhA* EF: If (you) come?	V+PP	আহা যদি ! TF: *aAhA yadi*
PP+PN	তেনেহলে সি ! TF: *tenehle si* EF: Or else he!	PN+PP	সি তেনেহলে ! TF: *si tenehale*

Assamese has a number of morpho-syntactic characteristics which makes it different from other Indic language such as Hindi. Our study reveals that - word order at the clause level is free, and in some cases intra clause level ordering is also free that is elements which can be thought as a single semantics unit, can be reorder within the clause. The most favourite word order of Assamese is SOV. For example-

1) মই ভাত খালোঁ। (SOV)
   TF: *mai bhAt khAlo.*

EF: I ate rice.
Now we can arrange these 3 constituents in 3! Ways. Thus we get 6 possible combinations.

a) ভাত মই খালোঁ। (OSV) *bhAt mai khAlo.*
b) ভাত খালোঁ মই। (OVS) *bhAt khAlo mai.*
c) মই খালোঁ ভাত। (SVO) *mai khAlo bhAt.*
d) খালোঁ ভাত মই। (VOS) *khAlo bhAt mai.*
e) খালোঁ মই ভাত। (VSO) *khAlo mai bhAt.*

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

2) মই তেজপুৰ বিশ্ববিদ্যালয়ৰ ছাত্ৰ। (PN-N-N)
TF: *maI Tezpur-ViswavidyAlayor chAtra.*
ET: I am student of Tezpur University
In this case the verb হয় (equivalent to " is " in English) is absent and is a meaningful sentence. Though there are 4 words, তেজপুৰ বিশ্ববিদ্যালয় (ৰ) is a single constituent, a name of an university so number of constituent will be 3 and hence total of 3! grammatically correct combinations are possible. Let us consider another sentence-

3) মানুহজনে কুকুৰটো ৰাস্তাত দেখিছে।
TF: *mAnuhjane kukurTo rAstAt dekhise.*
ET: The man has seen the dog on the road.

NP—মানুহজনে (the man) (Man + Qnt: Single + Gender: Male + Vibhakti)
NP—কুকুৰটো (the dog) (dog + Qnt:Single + Gender: Neuter)
NP—ৰাস্তাত (on road) (road + Vibhakti)
VP—দেখিছে (saw) (see + past ind.)
Interesting property of such type of sentence is that we can simply exchange the position of noun phrase (NP) without changing the emphasis.

a) কুকুৰটো মানুহজনে ৰাস্তাত দেখিছে।
TF: *kukurTo mAnuhjane rAstat dekhise*
b) ৰাস্তাত মানুহজনে কুকুৰটো দেখিছে।
TF: *rAstAt mAnuhjane kukurTo dekhise*

If we put a numeral classifier এটা before NP কুকৰ then total number of constituent will be increased to 5, and the sentence will be-

4) মানুহজনে এটা কুকুৰ ৰাস্তাত দেখিছে।
TF: *mAnuhjane etA kukur rastat dekhise.*
EF: The man saw a dog on road.
In this case we will not get 5! numbers of grammatically correct combination. Because the count noun এটা(*etA*) modifies only কুকুৰ(*kukur*), not the others. Therefore during reordering of a sentence এটা কুকুৰ(*etA kukur*) is considered as a single constituent. Sometime within the constituent reordering of words are also possible. For example- এটা কুকুৰ(*etA kukur*) can be written as কুকুৰ এটা(*kukur etA*) without changing he meaning of the phrase. But from the sentence it will not be clear whether "The man saw a dog on road" or "The man saw dog on a road".

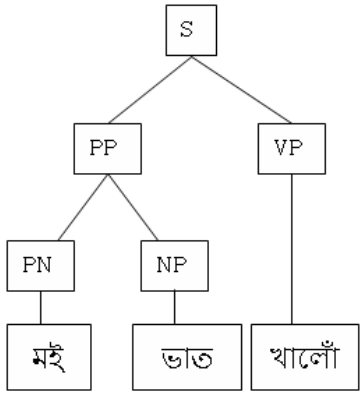


Fig. 1. Parse tree for sentence 1

a) মানুহজনে কুকুৰ এটা ৰাস্তাত দেখিছে।
TF: *mAnuhjane kukur etA rAstAt dekhise.*

5) আম মিঠা ফল।(N-ADJ-N)
TF: *aAm mithA phal.*
EF: Mango is fruit.
Here in this simple 3 constituent sentence if we try to exchange the position of noun(like example sentence 4) then we will get struturally correct but semantically wrong sentence.

a) ফল মিঠা আম.
TF: *phal mitha aAm*

Another important rule in this context is that the extension (Ext.) or the clauses as Ext. are always preceded by or followed by the constituent qualified. That is if element *A* is extension of *B* then *B* must be followed by *A* (in other words *A* does not occur after *B*). Consider the following example-

6) প্ৰধান শিক্ষকে আমাক সুন্দৰকৈ নতুন ব্যাকৰণ শিকাইছে ।
TF: *pradhAn sikhyake aAmak sundarkoi natun vyAkaran sikAIse*
EF: Head sir teaches us new grammar nicely.

প্ৰধান_Adj শিক্ষকে_N আমাক_PN সুন্দৰকৈ_Adv নতুন_Adj ব্যাকৰণ_N শিকাইছে_V

V. Parsing Assamese Sentences

As an initial exercise in parsing Assamese sentences, we present an approach for parsing simple sentences. We define a CFG grammar through which we can parse simple sentences like sentence (1) or any types of simple sentence where object is prior to verb. The parse tree of sentence (1) using the defined CFG grammar is shown in figure 1. In case of sentences 1(d) and 1(e) it generates a cross parse tree (Figure 2).

But unfortunately it can also generate a parse tree for sentence 5(a), which is semantically wrong. From sentence number 4 and 5 we can draw a conclusion that if the noun is attached with any type suffix, then it is easy for the defined CFG grammar to generate syntactically and semantically correct parse tree.

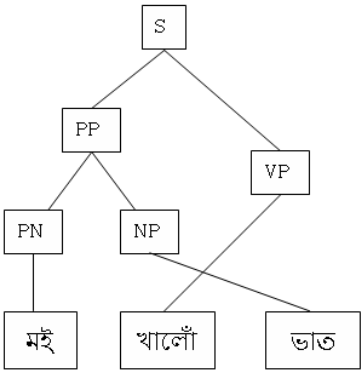


Fig. 2. Parse tree for sentence 1(d)

In Assamese two basic types of groupings of words are possible in a sentence. One is grouping adverb with verb and other is grouping adjective with noun. In general adverb or adjective occurs before the verb or noun respectively. Since Assamese is a relatively free word order language so these modifiers may occur anywhere in the sentence prior to verb or noun. It means that some constituent may occur in between adverb and verb or adjective and noun. In example sentence number 6, three types of grouping are possible- one verb group and two noun groups. Adjectives are adjacent to nouns but adverb occur prior to verb with a noun group in between. So after grouping we will get total 4 groups (Figure 3).

Fig. 3. Grouping of words of sentence 6

So we will get 4! grammatically correct sentences. But interestingly the main sentence from which the groups are formed is not included in this 4! combination. That is reordering the adverb again we can get another 6 new combinations. Though we mentioned above that adverb always occurs prior to verb, it is not always true. For example we can change the position of adverb and and verb within the group. That is সুন্দৰকৈ শিকাইছে can be reordered as শিকাইছে সুন্দৰকৈ. We can exchange the position of main object and subordinate object also. The constituent প্রধান শিক্ষক can be changed to শিক্ষক প্রধান. But here symbol of *Prathama Vibhakti* (Nominative case marker) এ is remove from S শিক্ষকে, and to the added to the Ext. of S. That is the new group will become শিক্ষক প্রধানে.

From figure 3 we can draw a complete graph considering each group as a vertex or node (Figure 4). A complete graph is a graph with all nodes are connected to each other. Now applying Chu-Liu-Edmond's maximum spanning tree algorithm we will obtain the parse tree for sentences which can not be obtained using our CFG grammar.

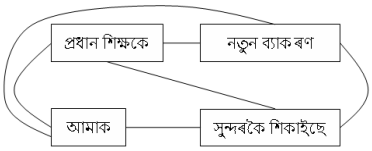


Fig. 4. Complete word graph

VI. Conclusion

Here we present the first step toward parsing of Assamese language. Our work is significant since Assamese has not received much attention of computational linguistic investigation. Using our approach we can handle simple sentences with multiple noun, adjective and adverb clauses. Handling of conjunction has been tackled to a limited extent. It needs to improved for complex sentences with different form. Also, there are other issues that we did not address in this paper.

References

[1] L. Tesniére, *Éléments de syntaxe structurelle*, Paris, Klincksieck, 1959.
[2] J. Bresnan and R. Kaplan, "Lexical-functional grammar: A formal system for grammatical representation," in *The Mental Representation of Grammatical Relations*, J. Bresnan, Ed., Cambridge, Massachusetts, 1982, MIT Press.
[3] A. K. Joshi, "An introduction to Tree Adjoining Grammar," *Mathematics of Language*, 1987.
[4] Carl Jesse Pollard and Ivan A. Sag, *Head-driven Phrase Structure Grammar*, University of Chicago Press, 1994.
[5] Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal, *Natural Language Processing: A Paninian Perspective*, Prentice-Hall, India, 1993.
[6] Adwait Ratnaparkhi, Salim Roukos, and R. Todd Ward, "A maximum entropy model for parsing," in *In Proceedings of the International Conference on Spoken Language Processing*, 1994, pp. 803–806.
[7] Michael A. Covington, "A dependency parser for variable-word-order languages," Tech. Rep., The University of Georgia, 1990.
[8] Akshar Bharati, Samar Husain, Dipti Misra Sharma, and Rajeev Sangal, "A two-stage constraint based dependency parser for free word order languages," in *Proceedings of the COLIPS International Conference on Asian Language Processing 2008 (IALP*, Chiang Mai, Thailand, 2008.
[9] Terry Koo, Xavier Carreras, and Michael Collins, "Simple semi-supervised dependency parsing," in *Proceedings of ACL / HLT*, 2008.
[10] Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret, "The conll 2007 shared task on dependency parsing," in *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, Prague, June 2007, p. 915932, Association for Computational Linguistics.
[11] Kenji Sagae and Alon Lavie, "A classifier-based parser with linear run-time complexity," in *Proceedings of the Ninth International Workshop on Parsing Technologies(IWPT),*. 2005, pp. 125–132, Association for Computational Linguistics.
[12] Joakim Nivre and Mario Scholz, "Deterministic dependency parsing of English text," in *Proceedings of COLING 2004*, Geneva, Switzerland, 2004, pp. 64–70.
[13] Hiroyasu Yamada and Yuji Matsumoto, "Statistical dependency analysis with support vector machines," in *Proceedings of the Ninth International Workshop on Parsing Technology*, 2003.
[14] Joakim Nivre and Jens Nilsson, "Pseudo-projective dependency parsing," in *Proceedings of the 43rd Annual Meeting of the ACL*, Ann Arbor, June 2005, pp. 99–106, Association for Computational Linguistics.
[15] Joakim Nivre, Johan Hall, and Jens Nilsson, "Memory-based dependency parsing," in *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL)*, Boston, Massachusetts, 2004, pp. 49–56.

[16] Yuchang Cheng, Masayuki Asahara, and Yuji Matsumoto, "Machine learning-based dependency analyser for Chinese," in *Proceedings of the International Conference on Chinese Computing (ICCC)*, 2005.
[17] Svetoslav Marinov and Joakim Nivre, "A data-driven dependency parser for Bulgarian," in *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories*, Barcelona, 2005, pp. 89–100.
[18] Gülşen Eryiğit and Kemal Oflazer, "Statistical dependency parsing of Turkish," in *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Trento, 3-7 April 2006, pp. 89–96.
[19] Michael Collins, *Head-Driven Statistical Models for Natural Language Parsing*, Ph.D. thesis, 1999.
[20] Michael Collins, Lance Ramshaw, and Jan Hajič, "A statical parser for Czech," in *Proceedings of the 37th Annual Meeting - Association for Computational Linguistics*, 1999, pp. 505–512.
[21] Brooke Cowan and Michael Collins, "Morphology and reranking for the statistical parsing of spanish," in *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Morristown, NJ, USA, 2005, pp. 795–802, Association for Computational Linguistics.
[22] Daniel M. Bikel, "Design of a multi-lingual, parallel-processing statistical parsing engine," in *Proceedings of the second international conference on Human Language Technology Research*, San Diego, california, 2002, pp. 178 –182, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.
[23] Amit Dubey and Frank Keller, "Probabilistic parsing for german using sister-head dependencies," July 2003.
[24] Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič, "Non-projective dependency parsing using spanning tree algorithms," in *Human Language Technologies and Empirical Methods in Natural Language Processing (HLT-EMNLP)*, 2005.
[25] Ryan McDonald and Fernando Pereira, "Online learning of approximate dependency parsing algorithms," in *Proc. EACL-06*, 2006.
[26] Jason M. Eisner, "Three new probabilistic model for dependency parsing: An exploration," in *Proceedings of COLING-96*, 1996.
[27] Gosse Bouma and Gertjan van Noord, "Constraint-based categorial grammar," in *Annual Meeting - Association for Computational Linguistics*, 1994.
[28] Beryl Hoffman, "A CCG approach to free word order language," in *Proceedings of 30th annual meeting of ACL'02*, 1992.

Named Entity Recognition: A Survey for the Indian Languages

Padmaja Sharma
Dept. of CSE
Tezpur University
Assam, India 784028
psharma@tezu.ernet.in

Utpal Sharma
Dept.of CSE
Tezpur University
Assam, India 784028
utpal@tezu.ernet.in

Jugal Kalita
Dept.of CS
University of Colorado at Colorado Springs
Colorado, USA 80918
kalita@eas.uccs.edu

***Abstract*—Named Entity Recognition(NER) is the process of identifying and classifying all proper noun into pre-defined classes such as persons, locations, organization and others. Work on NER in Indian languages is a difficult and challenging task and also limited due to scarcity of resources, but it has started to appear recently. In this paper we present a brief overview of NER and its issues in the Indian languages. We also describe the different approaches used in NER and also the work in NER in different Indian languages like Bengali, Telugu, Hindi, Oriya and Urdu along with the methodologies used. Lastly we presented the results obtained for the different Indian languages in terms of F-measure.**

I. Introduction

Natural Language Processing (NLP) is the computerized approach for analyzing text that is based on both a set of theories and a set of technologies. Named Entity Recognition (NER) is an important task in almost all NLP areas such as Machine Translation (MT), Question Answering (QA), Automatic Summarization (AS), Information Retrieval(IR), Information Extraction(IE), etc.

NER can be defined as a two stage problem - Identification of the proper noun and the classification of these proper noun into a set of classes such as person names, location names (cities, countries etc), organization names (companies, government organizations, committees, etc.), miscellaneous names (date, time, number, percentage, monetary expressions, number expressions and measurement expressions). Thus NER can be said as the process of identifying and classifying the tokens into the above predefined classes.

II. Basic Problems in Named Entity Recognition

The basic problems of NER are-

1) Common noun Vs proper noun- Common noun sometimes occurs as a person name such as “Suraj” which means sun, thus creating ambiguities between common noun and proper noun.
2) Organization Vs person name- “Amulya” as a person name as well as an organization, that creates ambiguity between proper noun and group indicative noun.
3) Organization Vs place name- “Tezpur” which act both as an organization and place name.
4) Person name Vs place name- When is the word “Kashi” being used as a person name and when as the name of a place.

Two broadly used approaches in NER are:

1) Rule-based NER
2) Statistics-based NER

Statistical methods such as Hidden Markov Model (HMM) [1], Conditional Random Field (CRF) [2], Support Vector Machine (SVM) [3], Maximum Entropy (ME) [4], Decision Tree (DT) [5] are the most widely used approaches. Besides the above two approaches, NER also make use of the Hybrid model which combines the strongest point from both the Rule based and statistical methods. This method is particularly used when data is less and complex Named Entities (NE) classes are used. Sirhari et.al [6] introduce a Hybrid system by combination of HMM, ME and handcrafted grammatical rules to build an NER system.

III. Problem faced in Indian Languages(ILs)

While significant work has been done in English NER, with a good level of accuracy, work in IL has started to appear only very recently. Some issues faced in Indian languages-

1) There is no concept of capitalization of leading characters of names in Indian Languages unlike English and other European languages which plays an important role in identifying NE’s.
2) Indian languages are relatively free-order languages.
3) Unavailability of resources such as Parts of speech (POS) tagger, good morphological analyzer, etc for ILs. Name lists are found available in web which are in English but no such lists for Indian Languages can be seen.
4) Some of the Indian languages like Assamese, Telugu are agglutinative in nature.
5) Indian languages are highly inflectional and morphologically rich in nature.

IV. Methodologies/Approaches

NER system can either be Rule-based or Statistics based. Machine Learning techniques(MLT)/Statistics based methods described below are successfully used for NER .

A. Hidden Markov Model (HMM):

HMM is a statistical model in which the system being modeled is assumed to be a Markov process with unobserved state. In this approach the state is not directly visible, but output depends on the state and is visible. Instead of single independent decisions, the model considers a sequence of decisions. Following are the assumptions of HMM-

- Each state depends on its immediate predecessor.
- Each observation value depends on the current state.
- Need to enumerate all observations.

.

The equation for HMM is given as-
$P(X) = \sum \prod_{i=0}^{n} P(y_i(y_{i-1})p(x_i|y_i))$

where,

$$X = (x_1, ..., x_n)$$

$$Y = (y_1, ..., y_n)$$

B. Conditional Random Field (CRF):

CRF are undirected graphical models a special case of which corresponds to conditionally trained finite state machines. They can incorporate a large number of arbitrary, non independent features and is used to calculate the conditional probability of values on designated output nodes given values on other designated input nodes. The conditional probability of a state sequence $S = (s1, s2..sT)$ given an observation sequence $O = (o1, o2, o3...ot)$ is calculated as

$$P(s|o) = \frac{1}{Z_o} \exp(\sum_{t=1}^{T} \sum_{k} \lambda_k f_k(S_{t-1}, S_t, o, t)$$

Where Z_o is a normalization factor overall state sequence.

$$Z_o = \sum \exp(\sum_{t=1}^{T} \sum_{k} \lambda_k f_k(S_{t-1}, S_t, o, t)$$

and $f_k(S_{t-1}, S_t, o, t)$ is a feature function whose weight λ_k is to be learned via training.

C. Support Vector Machine(SVM):

SVM first introduced by Vapnik are relatively new machine learning approaches for solving two-class pattern recognition problem. In the field of NLP, SVM is applied to text categorization and are reported to have high accuracy. It is a supervised machine learning algorithm for binary classification.

D. Maximum Entropy (ME):

The Maximum Entropy framework estimates probabilities based on the principle of making as few assumptions as possible other than the constraints imposed. Such constraints are derived from training data, expressing some relationship between features and outcomes. The probability distribution that satisfies the above property is the one with the highest entropy and has the exponential form

$P(o|h) = \frac{1}{z(h)} \prod_{j=1}^{k} \propto_j f_j(h, o)$

where o refers to the outcome, h the history(or context) and z(h) is a normalization function. In addition each feature function $f_j(h, o)$ is a binary function. The parameter $\propto_j$ are estimated by a procedure called Generalized Iterative Scaling(GIS) [7]. This is an iterative method that improves the estimation of the parameter at each iteration.

E. Decision Tree (DT):

DT is a powerful and popular tool for classification and prediction. The attractiveness of DT is due to the fact that in contrast to neural network, it represents rules. Rules can readily be expressed so that human can understand them or even directly use them in a database access language like SQL so that records falling into a particular category may be tree.

Decision Tree is a classifier in the form of a tree structure where each node is either a leaf node-indicates the value of the target attributes(class)of expressions, or a decision node that specifies some test to be carried out on a single attribute value with one branch and sub-tree for each possible outcome of the text. It is an inductive approach to acquire knowledge on classification.

V. Existing work on different Indian languages in NER

A. Hindi

Saha et.al(2008) [8] describes the development of Hindi NER using ME approach. The training data consists about 234 k words,collected from the newspaper "Dainik Jagaran" and is manually tagged with 17 classes including one class for not name and consists of 16,482 NEs. The paper also reports the development of a module for semi-automatic learning of context pattern. The system was evaluated using a blind test corpus of 25K words having 4 classes and achieved an F-measure of 81.52%.

Goyal(2008) [9] focuses on building a NER for Hindi using CRF. This method was evaluated on test set1 and test set 2 and attains a maximum F1-measure around 49.2% and nested F1-measure around 50.1% for test set1 maximum F1-measure around 44.97% and nested F1-measure around 43.70% for test set2 and F-measure of 58.85% on development set.

Saha et.al(2008) [10] has identified suitable features for Hindi NER task that are used to develop an ME based Hindi NER system. Two-phase transliteration methodology was used to make the English lists useful in the Hindi NER task. The system showed a considerable performance after using the transliteration based gazetteer lists. This transliteration approach is also applied to Bengali besides Hindi NER task and is seen to be effective. The highest F-measure achieved by ME based system is 75.89% which is then inceased 81.2% by using the transliteration based gazetteer list.

Li and McCallum(2004) [11] describes the application of CRF with feature induction to a Hindi NER. They discover

relevant features by providing a large array of lexical test and using feature induction to construct the features that increases the conditional likelihood. Combination of Gaussian prior and early-stopping based on the results of 10-fold cross validation is used to reduce over fitting.

Gupta and Arora(2009) [12] describes the observation made from the experiment conducted on CRF model for developing Hindi NER. It shows some features which makes the development of NER system complex. It also describes the different approaches for NER. The data used for the training of the model was taken from Tourism domain and it is manually tagged in IOB format.

B. Bengali

It is the seventh popular language in the world, second in India and the national language of Bangladesh. Ekbal and Bandyopadhyay(2009) [13] reports about the development of NER in Bengali by combining the output of the classifier like ME, CRF and SVM. The training set consists of 150k word form to detect the four Named Entity tags namely person, location, organization and miscellaneous names. Lexical context pattern generated from an unlabeled Bengali corpus containing 3 million wordform have been used to improve the performance of the classifier. Evaluation results of 30K wordforms have found the overall recall, precision and f-score values of 87.11%, 83.61% and 85.32%, which shows an improvement of 4.66% in f-score over the best performing SVM based system and an improvement of 9.5% in f-score over the least performing ME based system.

On the other hand work by Ekbal et.al [14] shows the development of Bengali NER system using the statistical CRF. The system make use of different contextual information of the words along with the variety of features for identifying Named Entity classes. The training set comprises of 150k wordform which is manually annotated with 17 tags. Experimental results of the 10-fold cross validation test shows the effectiveness of proposed CRF based NER system with an overall average recall, precision and f-score values of 93.8%, 87.8% and 90.7%.

Ekbal and Bandyopadhyay(2010) [15] developed NER system for Hindi and Bengali using SVM. An annotated corpora of 122,467 tokens of Bengali and 502,974 tokens of Hindi has been used tagged with 12 NE classes. The NER system has been tested with the gold standard test sets of 35K, and 60K tokens for Bengali and Hindi. Evaluation results have demonstrated the recall, precision and f-score of 88.61%, 80.12% and 84.15% for Bengali whereas 80.23%, 74.34% and 77.17% for Hindi.

Hasan et.al(2009) [16] presented a learning-based named entity recognizer for Bengali that donot rely on manually-constructed gazetteers in which they developed two architectures for the NER system. The corpus consisting of 77942 words is tagged with one of 26 tags in the tagset defined by IIT Hyderabad where they used CRF++ to train the POS tagging model. Evaluation results shows that the recognizer achieved an improvement of 7.5% in F-measure over a baseline recognizer.

Chaudhuri and Bhattacharya(2008) [17] has made an experiment on automatic detection of Named Entities in Bangla. Three-stage approach has been used namely-dictionary based for named entity, rules for named entity and left-right co-occurrences statistics. Corpus of Anandabazar Patrika has been used from the year 2001-2004. The manual tagging was done by the linguistic based on the global knowledge. Experimental results has shown the average recall, precision and f-measure to be 85.50%,94.24% and 89.51%.

Ekbal and Bandyopadhyay(2008) [18] developed NER system for Bengali using SVM. The system makes use of the different contextual information of the words along with the variety of features that are helpful in predicting the Named entities. A partially NE tagged Bengali news corpus has been used to create the training set for the experiment and the training set consists of 150K wordforms that is manually tagged with 17 tags. Experimental results of the 10 fold cross validation test shows the effectiveness of the proposed SVM based NER system with the overall average recall, precision and F-score values of 94.3%, 89.4% and 91.8%.

Ekbal and Bandyopadhyay(2008) [19] reports about the development of Bengali news corpus from the web consisting of 34 million wordforms. A part of this corpus of 150K wordforms is manually tagged with 16 NE and one non NE tag and additionally 30 K wordforms is tagged with a tagset of 12 NE tags defined for the IJCNLP-08 NER shared task for SSEAL. A tag conversion routine has been developed to convert the 16 NE tagged corpus of 150 K wordforms to the corpus tagged with IJCNLP-08 12 NE tags where the former has been used to develop the Bengali NER system using HMM, ME,CRF, SVM. Evaluation results of the 10 fold cross validation tests gives the F-score of 84.5% for HMM, 87.4% for ME, 90.7% for CRF and 91.8% for SVM.

Ekbal and Bandyopadhyay(2008) [20]describes the development of a web-based Bengali news corpus consisting of 34 million wordforms.The performance of the system is compared for two system- one is by using the lexical contextual patterns and the other using linguistic features along with the same set of lexical contextual pattern and came with the conclusion that the use of linguistic knowledge yields an highest F-value of 75.40%, 72.30%, 71.37% and 70.13% for person, location, organization and miscellaneous names.

Ekbal and Bandyopadhyay(2009) [21] describes a voted NER system by using Appropriate Unlabeled Data. This method is based on supervised classifier namely ME, SVM, CRF where SVM uses two different system known as forward parsing and backward parsing. The system has been tested for Bengali containing 35,143 news document and 10 million wordfroms and makes use of language independent features along with different contextual information of the words. Finally the models have been combined together into a final system by a weighted voting technique and the experimental

results show the effectiveness of the proposed approach with the overall recall precision and f-score values of 93.81%, 92.18% and 92.98%.

Ekbal and Bandyopadhyay(2008) [22] reports about the development of NER system in Bengali by combining the outputs of the classifier like ME, CRF, SVM. The corpus consisting of 250K wordforms is manually tagged with four NE namely person, location, organization and miscellaneous. The system makes use of different contextual information of the words along with the variety of features that helps in identifying the NEs. Experimental results shows the effectiveness of the proposed approach with the overall average recall, precision and f-score values of 90.78%, 87.35% and 89.03% respectively. This shows an improvement of 11.8% in f-score over the best performing SVM based baseline system and an improvement of 15.11% in f-score over the least performing ME based baseline system.

Hasanuzzaman et.al(2009) [23] describes the development of NER system in Bengali and Hindi using ME framework with 12 NE tags. A tag conversion routine has been developed in order to convert the fine-grained NE tagset of 12 tags to a coarse-grained NE tagset of 4 tags namely person name, location name, organization name and miscellaneous name. The system makes use of different contextual information of the words along with the variety of orthographic word - level features that helps in predicting the four NE classes. Ten fold cross validation test results the average recall, precision and f-measure of 88.01%, 82.63%, 85.22% for Bengali and 86.4%, 79.23% and 82.66% for Hindi.

Ekbal and Bandyopadhyay(2007) [24] reported the development of HMM based NER system. For Bengali it was tested manually over a corpus containing 34 million wordforms developed from the online Bengali newspaper. A portion of the tagged news corpus containing 150,000 wordforms is used to train the NER system through HMM-based parts of speech tagger with 26 different POS tags and the training set thus obtained is a corpus tagged with 16 NE tags and one non NE tag and the experimental results of the 10-fold cross validation yields an average Recall, Precision and F-score values of 90.2%, 79.48% and 84.5% respectively. After this the HMM-based NER system is also trained and tested with Hindi data to show the effectiveness for the language independent features. Tthe results for Hindi NER shows an average Recall, Precision and F-score values of 82.5%, 74.6% and 78.35% respectively.

C. Telugu

Telugu being a language of the Dravidian family, is the third most spoken language in India and official language of Andhra Pradesh.

Srikanth and Murthy (2008) [25] have used part of the LERC-UoH Telugu corpus where CRF based Noun Tagger is built using 13,425 words manually tagged data and tested on a test data set of 6,223 words and came out with an F-measure of 91.95%. Then they develop a rule-based NER system consisting of 72,152 words including 6,268 Named Entities where they identified some issues related to Telegu NER and later develop a CRF based NER system for telegu and obtained an overall F-measures between 80% and 97% in various experiments.

Shishtla et.al(2008) [26] conducted an experiment on the development data released as a part of NER for South and South East Asian Languages (NERSSEAL) Competition. The Corpus consisting of 64026 tokens was tagged using the IOB format (Ramshaw and Marcus, 1995). The author have showed experiments with various features for Telugu. The best performing model gave an F-1 measure of 44.91%.

Raju et.al [27] have developed a Telugu NER system by using ME approach. The corpus was collected from the iinaaDu, vaarta news papers and Telugu Wikipedia. Manually tagged test data is prepared to evaluate the system. The system makes use of the different contextual information of the words and Gazetteer list was also prepared manually or semi-automatically from the corpus and came out with a an F-measure of 72.07% for person, 6.76%, 68.40% and 45.28% for organization, location and others respectively.

D. Tamil

VijayKrishna and Sobha(2008) [28] developed a domain specific Tamil NER for tourism by using CRF. It handles morphological inflection and nested tagging of named entities with a heirarchial tageset consisting of 106 tags. A corpus of 94k is manually tagged for POS, NP chunking, and NE annotations. The corpus is divided into training data and the test data where CRF is trained with the former one and CRF models for each of the levels in the hierarchy are obtained. The system comes out with a F-measure of 80.44%.

Pandian et.al(2008) [29] presented a hybrid three-stage approach for Tamil NER. The E-M(HMM) algorithm is used to identify the best sequence for the first two phases and then modified to resolve the free-word order problem. Both NER tags and POS tags are used as the hidden variables in the algorithm. Finally the system comes out with an F-measure of about 72.72% for various entity types.

E. Oriya

Biswas et.al [30] presented a hybrid system for Oriya NER that applies both ME and HMM and some handcrafted rules to recognize NEs. Firstly the ME model is used to identify the named entities from the corpus and then this tagged corpus is regarded as training data for HMM which is used for the final tagging. Different features have been considered and linguistic rules help a lot for identification of named entities. The annotated data used in the system is in IOB format. Finally the system comes with an F-measure between 75% to 90%.

VI. Analysis

From the above survey we have seen that though the work in NER in IL is limited, still considerable work has been done for the Bengali language. The level of accuracy obtained for these languages are described in the (Table 1, 2) along with the approaches used. We can see that CRF is the most widely used approach which shows an effective results for the Indian Languages in comparison to the other approaches. Our survey reveals that Ekbal and Bandyopadhyay [18] achieved highest accuracy using CRF 90.7%, using SVM 91.8, using ME 87.4% and using HMM 84.5% for Bengali.

VII. Conclusion and Future Work

In this survey we have studied the different techniques employed for NER, and have identified the various problems in the task particularly for ILs. In addition to these approaches researchers can also try using other approaches like DT, Genetic algorithm, Artificial and Neural Network etc that which already showed an excellent performance in the other languages like English, Germany etc. Also NER should be attempted for other IL in which no such work has been attempted so far.

TABLE I
Comparison of the approaches with their accuracy for the different Indian Languages. Fm : Maximal F-measure, Fn : Nested F-Measure, Fl: Lexical F-measure, BIA : Baseline Induced Affixes, BIAW : Baseline Induced Affixes Wiki: Classifier- outputs of ME, CRF,SVM.

Language	Author	Approach	Accuracy(%)
Telugu	[25]	CRF	80-97
	[26]	CRF	44.91
	[27]	ME	P-72.07
			O-60.76
			L-68.40
			Others-45.28
Tamil	[28]	CRF	80.44
	[29]	HMM	72.72
Hindi	[10]	ME	75.89
	[8]	ME	81.52
	[9]	CRF	58.85
Bengali	[18]	SVM	91.8
	[17]	n-gram	89.51
	[14]	CRF	90.7
	[13]	Classifiers	85.32
	[19]	MLT	HMM- 84.5
			ME -87.4
			CRF -90.7
			SVM -91.8
	[21]	Classifier	92.98
	[22]	Classifier	89.03
	[20]	MLT	P-75.40
			L-72.30
			O-71.37
			Others-70.13
	[16]	CRF	Baseline-65.57
			BIA-69.32
			BIAW-71.99
Bengali+Hindi	[15]	SVM	Bengali-84.15
			Hindi-77.17
Bengali+Hindi	[23]	ME	Bengali-85.22
			Hindi-82.66
Bengali+Hindi	[24]	HMM	Bengali-84.5
			Hindi-78.35

TABLE II
Comparison of the approaches with their accuracy for South and South East Asian Languages

Author	Approach	Language	Fm	Fn	Fl	F measure
[31]	CRF	Bengali	53.36	53.46	59.39	-
[32]	ME	Hindi	-	-	-	65.13
		Bengali	-	-	-	65.96
		Oriya		-	-	44.65
		Telugu	-	-	-	18.74
		Urdu		-	-	35.47
[33]	CRF	Bengali	35.65	33.94	40.63	-
		Hindi	48.71	50.47	50.06	-
		Oriya	29.29	26.06	39.04	-
		Telugu	8.19	43.19	40.94	-
		Urdu	39.86	39.01	43.46-	-
[34]	ME	Bengali	12.50	11.97	12.30	-
		Hindi	29.24	28.48	25.68	-
		Oriya	13.94	11.91	19.44	-
		Telugu	00.32	01.08	08.75	-
		Urdu	26.41	24.39	27.73-	-
[35]	CRF	Bengali	31.48	30.79	35.71	-
		Hindi	42.27	41.56	40.49	-
		Oriya	25.66	22.82	36.76	-
		Telugu	21.56	17.02	45.62	-
		Urdu	33.17	31.78	38.25-	-
	HMM	Bengali	33.50	32.83	39.77	-
		Hindi	48.30	47.16	46.84	-
		Oriya	28.24	25.86	45.84	-
		Telugu	13.33	32.37	46.58	-
		Urdu	34.48	36.83	44.73-	-
[36]	N-gram	Telugu	-	-	-	49.62
		Hindi	-	-	-	45.07

References

[1] B. D. M, M. Scott, S. Richard, and W. Ralph, "A High Performance Learning Name-finder," in *Proceedings of the fifth Conference on Applied Natural language Processing*, 1997, pp. 194–201.

[2] J. Lafferty, A. McCallum, and F. Pereira, "Probabilistic Models forSegmenting and Labelling Sequence Data," in *Proceedings of the Eighteenth International Conference on Machine Learning(ICML-2001)*, 2001.

[3] Cortes and Vapnik, "Support Vector Network ,MachineLearning," 1995, pp. 273–297.

[4] B. Andrew, "A Maximum Entropy Approach to NER," Ph.D. dissertation, 1999.

[5] F.Bechet, A.Nasr, and F.Genet, "Tagging Unknown Proper Names using Decision Trees," in *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistic*, 2000.

[6] R.Sirhari, C.Nui, and W.Li, "A Hybrid Approach for Named Entity and Sub-Type Tagging," in *Proceedings of the sixth conference on Applied natural language processing, Acm Pp*, 2000, pp. 247–254.

[7] J. Darroch and D.Ratcliff, "Generalized iterative scaling for log-linear models," *The Annals of Mathematical Statistics*, vol. 43(5), 1972.

[8] S. K. Saha, S. Sarkar, and P. Mitra, "A Hybrid Feature Set based Maximum Entropy Hindi Named Entity Recognition," in *Proceedings of the 3rd International Joint Conference on NLP*, Hyderabad,India, January 2008, pp. 343–349.

[9] A. Goyal, "Named Entity Recognition for South Asian Languages," in *Proceedings of the IJCNLP-08 Workshop on NER for South and South-East Asian Languages*, Hyderabad, India, Jan 2008, pp. 89–96.

[10] S. K. Saha, P. S. Ghosh, S. Sarkar, and P. Mitra, "Named Entity Recognition in Hindi using Maximum Entropy and Transliteration," *Research journal on Computer Science and Computer Engineering with Applications*, pp. 33–41, 2008.

[11] W. Li and A. McCallum, "Rapid Development of Hindi Named Entity Recognition using Conditional Random Fields and Feature Induction (Short Paper)," *ACM Transactions on Computational Logic*, pp. 290–294, Sept 2003.

[12] P. K. Gupta and S. Arora, "An Approach for Named Entity Recognition System for Hindi: An Experimental Study," in *Proceedings of ASCNT-2009*, CDAC, Noida, India, pp. 103–108.

[13] A. Ekbal and S. Bandyopadhyay, "Bengali Named Entity Recognition using Classifier Combination," in *Proceedings of 2009 Seventh International Conference on Advances in Pattern Recognition*, pp. 259–262.
[14] A. Ekbal, R. Haque, and S. Bandyopadhyay, "Named Entity Recogntion in Bengali: A Conditional Random Field," in *Proceedings of ICON*, India, pp. 123–128.
[15] A. Ekbal and S. Bandyopadhyay, "Named Entity Recognition using Support Vector Machine: A Language Independent Approach," *International Journal of Computer, Systems Sciences and Engg(IJCSSE)*, vol. 4, pp. 155–170, 2008.
[16] K. S. Hasan, M. ur Rahman, and V. Ng, "Learning -Based Named Entity Recognition for Morphologically-Rich Resource-Scare Languages," in *Proceedings of the 12th Conference of the European Chapter of the ACL*, Athens, Greece, 2009, pp. 354–362.
[17] B. B. Chaudhuri and S. Bhattacharya, "An Experiment on Automatic Detection of Named Entities in Bangla," in *Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian laanguages*, Hyderabad, India, January 2008, pp. 75–82.
[18] A. Ekbal and S. Bandyopadhyay, "Bengali Named Entity Rcognition using Support Vector Machine," in *Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian laanguages*, Hyderabad, India, January 2008, pp. 51–58.
[19] ——, "Development of Bengali Named Entity Tagged Corpus and its Use in NER System," in *Proceedings of the 6th Workshop on Asian Language Resources*, 2008.
[20] ——, "A web-based Bengali news corpus for named entity recognition," *Language Resources & Evaluation*, vol. 42, pp. 173–182, 2008.
[21] ——, "Voted NER System using Appropriate Unlabelled Data," in *Proceedings of the 2009 Named Entities Workshop, ACL-IJCNLP 2009*, Suntec, Singapore, August 2009, pp. 202–210.
[22] ——, "Improving the Performance of a NER System by Post-processing andVoting," in *Proceedings of 2008 Joint IAPR International Workshop on Structural Syntactic and Statistical Pattern Recognition*, Orlando, Florida, 2008, pp. 831–841.
[23] M. Hasanuzzaman, A. Ekbal, and S. Bandyopadhyay, "Maximum Entropy Approach for Named Entity Recognition in Bengali and Hindi," *International Journal of Recent Trends in Engineering*, vol. 1, May 2009.
[24] A. Ekbal and S. Bandyopadhyay, "AHidden Markov Model Based Named Entity Recognition System:Bengali and Hindi as Case Studies," in *Proceedings of 2nd International conference in Pattern Recognition and Machine Intelligence*, Kolkata, India, 2007, pp. 545–552.
[25] P.Srikanth and K. N. Murthy, "Named Entity Recognition for Telegu," in *Proceedings of the IJCNLP-08 Wokshop on NER for South and South East Asian languages*, Hyderabad, India, Jan 2008, pp. 41–50.
[26] P. M. Shishtla, K. Gali, P. Pingali, and V. Varma, "Experiments in Telegu NER: A Conditional Random Field Approach," in *Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian laanguages*, Hyderabad, India, January 2008, pp. 105–110.
[27] G. Raju, B.Srinivasu, D. S. V. Raju, and K. Kumar, "Named Entity Recognition for Telegu using Maximum Entropy Model," *Journal of Theoretical and Applied Information Technology*, vol. 3, pp. 125–130, 2010.
[28] V. R and S. L, "Domain focussed Named Entity Recognizer for Tamil using Conditional Random Fields," in *Proceedings of the IJCNLP-08 Wokshop on NER for South and South East Asian languages*, Hyderabad, India, 2008, pp. 59–66.
[29] S. Pandian, K. A. Pavithra, and T. Geetha, "Hybrid Three-stage Named Entity Recognizer for Tamil," *INFOS2008*, March 2008.
[30] S.Biswas, S.P.Mohanty, S.Acharya, and S.Mohanty, "A Hybrid Oriya Named Entity Recogntion system," in *Proceedings of the CoNLL*, Edmonton, Canada, 2003.
[31] A. Ekbal, R. Haque, A. Das, V. Poka, and S. Bandyopadhyay, "Language Independent Named Entity Recognition in Indian Languages," in *Proceedings of the IJCNLP-08 Wokshop on NER for South and South East Asian languages*, Hyderabad, India, 2008, pp. 33–40.
[32] S. K. Saha, S. Chatterji, and S. Dandapat, "A Hybrid Approach for Named Entity Recognition in Indian Languages," in *Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian laanguages*, Hyderabad, India, January 2008, pp. 17–24.
[33] K. Gali, H. Surana, A. Vaidya, P. Shishtla, and D. M. Sharma, "Aggregating Machine Learning and Rule Based Heuristic for Named Entity Recognition," in *Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian laanguages*, Hyderabad, India, January 2008, pp. 25–32.
[34] A. K. Singh, "Named Entity Rcognition for South and South East Asian Languages," in *Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian laanguages*, Hyderabad, India, January 2008, pp. 5–16.
[35] P. K. P and R. K. V, "A Hybrid Named Entity Rcognition System for South Asian Languages," in *Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian laanguages*, Hyderabad, India, January 2008, pp. 83–88.
[36] P. M. Shishtla, P. Pingali, and vasudeva Varma, "A Character n-gram Approach for Improved Recall in Indian Language NER," in *Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian laanguages*, Hyderabad, India, January 2008, pp. 67–74.

An Implementation of APERTIUM Morphological Analyzer and Generator for Tamil

Parameshwari K

CALTS, University of Hyderabad,
Hyderabad-500046.

parameshkrishnaa@gmail.com,

***Abstract*—** A Morphological Analyzer and Generator are two crucial tools involving any Natural Language Processing of Dravidian Languages. The present paper discusses the improvization of the existing Morphological Analyzer and Generator for Tamil by defining and describing the relevant linguistic database required for the purpose of developing them. The implementation of an open source platform called Apertium to handle inflection as well as derivation for word level analysis and generation of Tamil is also discussed. The paper also presents the efficacy, coverage and speed of the module against the large corpora. The paper also draws inferences of the morphological categories in their inflection and problems in analysing them.

I. Introduction

A language like Tamil is regarded as morphologically rich wherein the words are formed of one or more stems/roots plus one or more suffixes. So the complexity of morphology requires a more sophisticated morphological analyzer and generator. A morphological analyzer is a computational tool to analyze word forms into their roots along with their constituent functional elements. The morphological generator is the reverse process of an analyzer i.e. from a given root and functional elements, it generates the well-formed word forms.

The present attempt involves a practical adoption of lttoolbox for the Modern Standard Written Tamil in order to develop an improvised open source morphological analyzer and generator. The tool uses the computational algorithm called Finite State Transducers for one-pass analysis and generation, and the database is based on the morphological model called Word and Paradigm.

II. Implementation of Apertium (lttoolbox[1])

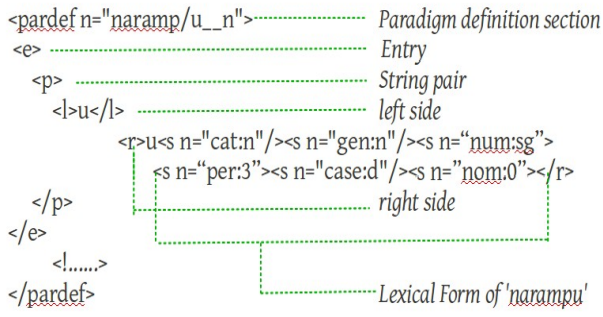
Apertium is an open source machine translation platform developed by the Transducens research group at the *Department de Llenguatges i Sistemes Inform`atics of the Universitat d'Alacant* in Spain. The lttoolbox is a toolbox for lexical processing such as morphological analysis and generation of words. The Document Type Definition (DTD) format is used in XML file for creating the lexical database in order to convert it into FST. The present attempt uses LINUX operating system with *fedora 10* platform for implementing the tool.

The analyzer as well as generator is obtained from a single morphological database, depending on the direction in which it is read by the system: read from left to right, we obtain the analyzer, and read from right to left, the generator.

The module requires the following database to build a Morphological Analyzer.

A. Paradigms and their definitions. A Paradigm here is referred to a complete set of related inflectional and productive derivational word forms of a given category. The database comprises of six distinct lexical categories viz., Noun, Verb, Adjective as open class and Pronoun, Number words and Locative Nouns as closed class. The Tamil Morphological Database available at the Centre for Applied Linguistics and Translation Studies (University of Hyderabad) Language Laboratory is extracted and improvised involving six distinct lexical inflectional categories for the purpose.

The Definition refers to the features and feature values of the root such as category, gender, number, person and case marking in the case of nouns and tense, aspect and modal category information in the case of verbs so on and so forth. The WX-notation[2] of transliteration is followed in this paper.



The XML format of inflection paradigm for Tamil 'narampu'

B. Linked Paradigms For Derivation. Derivational forms need the dynamic analysis rather than putting in the Dictionary. It is an alternative lexico-semantic modal which operates along

with inflection. There is a layer that introduces the lexemes into derivation and concurrently follows the inflection of the derived lexeme. For instance, *patikkirYavanY* 'one who(he) is reading' is a derived pronominal of the verb *pati* 'read'. It further takes all the inflections of pronoun 'avanY'. Here the derived pronoun is linked with the pronoun paradigm *avanY*.

```
<pardef n="pati__v">
 <e>
   <p>
     <l>kirYavanY</l>
       <r><s n="v"/><s n="m"/><s n="sg"/><s n="3"/><s n="0"/>
          <s n="kirY_a"/></r>
     </p><i>kirYavanY</i><par n="avanY__P"/>  ..... Linking paradigm for derivation
 </e>
    <!.....>
</pardef>
```

THE XML FORMAT OF PARADIGM TO HANDLE DERIVATION

C. LEXICON. A root word dictionary in Morphological Analyzer differs from a conventional dictionary. The dictionary for Morphological Analysis which is built for Word and Paradigm Model contains roots, categories and their corresponding paradigm. The present Morphological analyzer-generator lexicon contains the root/lemma, the part of the lemma which is common to all the inflected forms, that is, it contains the lemma cut at the point in which the paradigm regularity begins along with the appropriate paradigm and the paradigm name.

```
<e lm="maram">  ..... Element for Lemma
   <i>mara</i>  ..... The part of the Lemma
   <par n="mara/m__n"/>
 </e>           ..... Paradigm name
```

A DICTIONARY ENTRY OF THE LEXEME 'MARAM'

D. COMPILING AND PROCESSING. The data is compiled and processed by using the applications used in the lexical processing modules and tools (lttoolbox). The applications are responsible for compiling dictionaries into a compact and efficient representation (a class of finite-state transducers called augmented letter transducers) and processing the compiled data for the real time text.

The 'lt-comp' is the application responsible for compiling dictionaries used by Apertium into a compact and efficient representation.

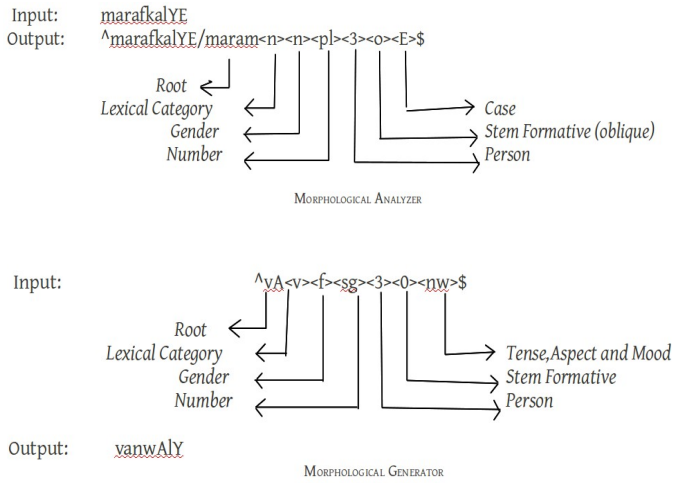
Synopsis **:** lt-comp [lr | rl] dictionary_file output_file

The dictionary which is compiled is processed by the application 'lt-proc' that is responsible for processing the data.

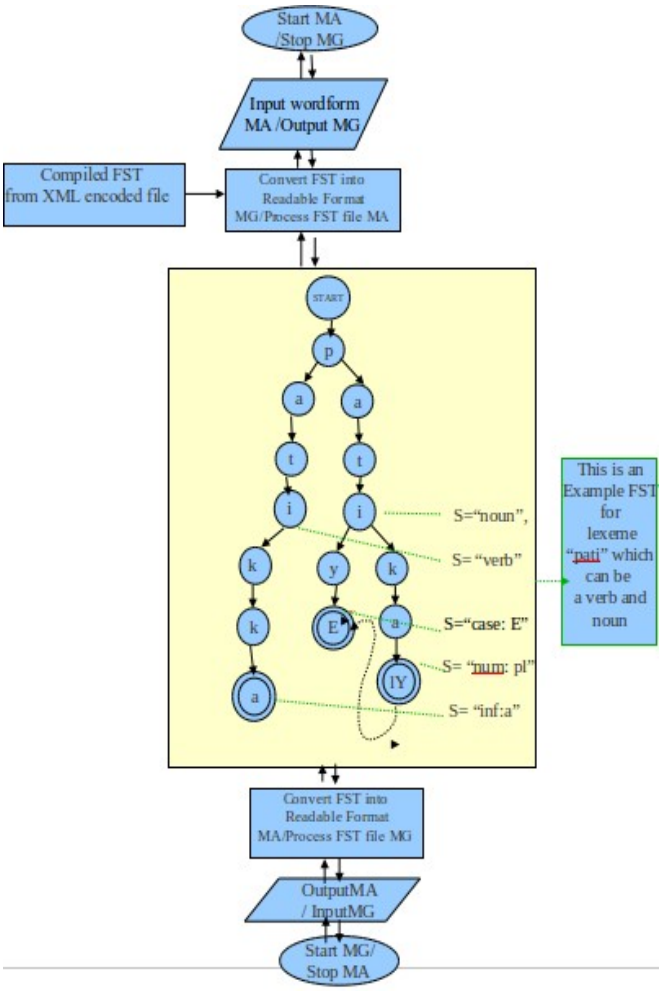
Synopsis **:** lt-proc [-c] [-a|-g] fst_file [input_file [output_file]]

The 'lt-proc' processes the stream with the letter transducers. Here 'fst_file' refers to the compilation file which is in FST format.

E. THE INPUT AND OUTPUT SPECIFICATION.



F. DATA FLOW IN MORPHOLOGICAL ANALYZER. The below figure is a flowchart that describes the data flow in the Morphological Analysis (MA) and Generation (MG).



G. Database. The following table shows the database of the Morphological module.

Paradigm			Dictionary Size (lemma) Number of Words	
Category	Number of Inflectional Classes	Number of Inflections per class	Category wise	Total
Noun	20	743	57,322	68,060
Verb	29	934	10,114	
Adjective	2	372	209	
Pronoun	11	654	18	
Numeral	14	370	129	
NST	7	67	62	
Avy	-	-	206	

Table 1 : Database

III. Testing and evaluation

The Morphological analyzer tool was tested with the corpus (CALTS corpus of 4.4 million words and EMILLI CIIL corpus of 4.8 million words) in order to find out it's coverage of the corpus. The coverage of the analyzer is calculated by dividing the analyzed word with the total number of words.

Corpus	Total words	Recognized words	Coverage	Speed
CALTS Corpus	4,45,130	3,75,891	84.44%	0m0.289s
EMILLI CIIL Corpus	4,85,543	4,05,898	83.59%	0m0.297s

The speed is an indication that CALTS-Apertium consumes less time to analyze a large number of data.

IV. Analysis

In the course of testing the tool, it has been found certain inconsistencies and lapses in recognizing certain words. The lapses are due to the lexical items with orthographic variation, inflectional variation, dialectal variation, naturalized loan words particularly from English into Tamil, proper nouns.

Type	Word From	Frequency in the Corpus
Orthographic Variation	*koyil* 'temple' *kovil* 'temple'	885 occurrences 204 occurrences
Inflectional Variation	*eVlYYuwwu-kkalY* 'letters' *eVlYYuwwu-kalY* 'letters'	57 occurrences 171 occurrences
Dialectal Variation	*vanwAy* 'You came'(standard) *vanweV* 'You came'(dialect)	765 occurrences 6 occurrences
Naturalized English loans	*polIS* 'police'	20070 occurrences
Proper nouns	*kaNNanY* 'male name' *wamilYYnAtu* 'Tamil Nadu'	211 occurrences 364 occurrences

The careful appraisal and study on the unrecognized words is conducted to identify and overcome the lapses by incorporating certain amount of data into the morphological database to enhance the coverage and the overall performance of the morphological tools. Other than these, the following problems are also well noted.

A. External sandhi. In Tamil, the obstruents (k,c,t,w,p) in the word initial position when preceded by a word form ending in a short vowel (a ,i, u, e, o), the diphthong (E), optionally glide y, ending in lYY and r appear as geminated and the first segment of which is always written as the final segment of the first word as shown below.

Examples for External Sandhi involves in Tamil.
*anwa**c** **c**attam* 'that law', *yAnYE**k** **k**utti* 'small elephant',
*curYrYulA**p** **p**ayaNi* 'tourists', *wAy**p** **p**Acam* 'motherly love',
*peVyar**p** **p**alakE* 'naming board', *wamilYY**w** **w**Ay* 'Mother of Tamil Nadu'.

However, the first words in each of these pairs is unrecognized because the additional word final consonant is the result of external sandhi. This requires the deletion of the consonants before they are passed on to the Morphological Analyzer.

B. Need for Sandhi Splitter. The words that are joined together require to be analyzed by Sandhi splitter beforehand. Or else, it will be a hectic task to add all the conjoined word forms in the database, since any subsequent words can be written together. The requirement of Sandhi Splitter is necessary to identify words which are combined together not due to inflectional rule. The sandhi splitter can separate these kinds of words which can be further forwarded to Morphological Analyzer. For instance,

nAteVfkum, nAtu + eVfkum 'nation+whole'
ifkulYlYa, ifku + ulYlYa 'here+being'
veNtumAnYAlum, veNtum+AnYAlum 'need+though'

C. Naturalized English Words. The words that are naturalized as Tamil especially from English need to be analyzed. The problem in identifying these words are a single word may have more than two orthographical and spelling variations. It differs according to the person how they pronounce. Therefore, it has to be studied through corpus that can reveal the different forms and their distributions.

For instance, for 'engineer'
inYginIr / eVnYginiyar / inYginYiyar

D. Colloquial Forms. In Tamil, the influence of colloquial forms can be normally seen in the written due to its nature of possessing two forms in Modern days as spoken and written. It is unavoidable to restrict the spoken, though it is informal. The problem may have been solved by providing the variant forms in the paradigmatic tables.

For instance,
porYAnY is used in spoken instead of *pokirYAnY* 'he is going'
paticcu for *patiwwu* 'having studied'

After implementing the above said suggestions, the analyzers may be expected to provide a more efficient and effective analysis.

V. Conclusion

The Apertium tool for Tamil is efficient in terms of time for processing a large number of words. The combination of Finite State Transducers (letter transducer) and the paradigm approach is more efficient and helps in faster parsing. The other advantage of the Apertium is that the current morphological database can be used to create a parallel morphological generator for Tamil.

[1] A finite state toolkit in Apertium to perform lexical processing

[2] Transliteration Scheme using wx-notation:
Tamil Orthography :
a A i I u U eV e E oV o O H
k f c F t N w n p m y r l v lYY lY rY nY j s h R

References

[1] Arden, A.H. 1976. *A progressive Grammar of the Tamil language*. Madras : The Christian Literature Society.

[2] Fercada, Mikel et.al. 2008. *Documentation of the Open-Source Shallow-Transfer Machine Translation platform Apertium.* Retrieved from http://www.gnu.org/copyleft/fd1.html.

[3] Parameswari K. 2009. *An improvised morphological Analyzer for Tamil: A case of implementing the opensource platform Apertium*. Unpublished M.Phil. Thesis. Hyderabad: University of Hyderabad.

[4] Ramaswamy, Vaishnavi. 2003. *A Morphological Analyzer for Tamil.* Unpublished Ph.D. Thesis. Hyderabad: University of Hyderabad.

[5] Uma Maheshwar Rao, G. Amba Kulkarni, P. and Christopher, M. 2007. *Morphological Analyzer and Its Functional Specifications for IL-ILMT System*. CALTS, Hyderabad: University of Hyderabad.

[6] Uma Maheshwar Rao, G. and Amba Kulkarni, P. 2006. *Computer Applications in Indian Languages*, Hyderabad: The centre for distance education, University of Hyderabad.

[7] Uma Maheshwar Rao, G. And Parameshwari, K. 2010. *On the Description of Morphological Data for Morphological Analysers and Generators: A case of Telugu, Tamil and Kannada.* Mona Parekh (ed.) in Morphological Analysers and Generators, pp73-81. Mysore:LDCIL,CIIL.www.ldcil.org/up/conferences/morph/presentation.html

[8] Uma Maheshwar Rao, G. and Christopher, M. 2010. *Word Synthesizer Engine.* Mona Parekh (ed.) in Morphological Analysers and Generators, pp73-81. Mysore: LDCIL,CIIL. www.ldcil.org/up/conferences/morph/presentation.html

[9] Uma Maheshwar Rao, G. 1999. *Morphological Analyzer for Telugu.* (electronic form). Hyderabad: University of Hyderabad.

[10] Uma Maheshwar Rao, G. 2002. *A Computational Grammar of Telugu*. (Momeograph) Hyderabad: University of Hyderabad.

[11] Vaidhya, Ashwini and Dipti Misra Sharma. 2009. *Using Paradigms for Certain Morphological phenomena in Marathi. 7th International Conference on NLP (ICON-2009).* New Delhi: Macmilan Publishers India Ltd.

[12] Viswanathan. S et.al. 2003. *A Tamil Morphological Analyser.* Recent Advances in NLP. 31-39. Mysore: Central Institute of Indian Languages.

ADVANCEMENT OF CLINICAL STEMMER

Pramod Premdas Sukhadeve[1] and Dr. Sanjay Kumar Dwivedi[2]

[1]Department of Computer Science, Babasaheb Bhimrao Ambedkar University (A Central University), Luck now, India
Sukhadeve.pramod@gmail.com
[2]Department of Computer Science, Babasaheb Bhimrao Ambedkar University (A Central University), Vidya Vihar Raebareli Road, Lucknow, India
Skd2000@yahoo.com

Abstract:

Word Stemming is common form of language processing in most Information Retrieval (IR) systems. Word stemming is an important feature supported by present day indexing and search systems. Idea is to improve by automatic handling of word endings by reducing the words to their word roots, at the time of indexing and searching. Stemming is usually done by removing any attached suffixes, and prefixes from index terms before the assignment of the term. Since the stem of a term represents a broader concept than the original term, the stemming process eventually increases the number of retrieved documents. Texts from the medical domain are an important task for natural language processing. This paper investigates the usefulness of a large medical database for the translation of medical documents using a rule based machine translation system. We are able to show that the extraction of affixes from the words.

Keywords: Stemming, Information Retrieval, Suffix, Prefix, Natural Language Processing.

Introduction:

Stemming is the procedure of finding the root word, by stripping away the affix attached to the word. In many languages words are often obtained by affixing existing words or roots. Stemming is a widespread form of language processing in most information retrieval systems [1]. It is similar to the morphological process used in natural language processing, but has somewhat different aims. In an Information retrieval system, stemming is used to reduce different word forms to common roots, and thereby improve the aptitude of the system to match query and document vocabulary. It also helps in clinical language to knob the clinical terms, names of deceases and symptoms of patient. Although stemming has been studied mainly for English, there is evidence that it is useful for a number of languages. Stemming in English is usually done during document indexing by removing word endings or suffixes using tables of common endings and heuristics about when it is appropriate to remove them. Thus using a stemmer improves the number of documents retrieved in response to translate the clinical data. Also, since many terms are mapped to one, stemming serves to decrease the size of the index files in the information retrieval system. Many stemming algorithms have been proposed, and there have been many experimental evaluations of these. But, very few work on stemming has been reported for clinical language. This paper investigates the usefulness of a large medical database for the translation of documents; we present a stemmer for clinical language. This conflates[1] terms by stripping off word endings from a suffix list maintained in a database.

---

[1]The term conflates is used to denote the act of mapping variants of a word to a single term or 'stem'.

English Stemming Word curriculum

The first step while developing a stemmer is to define the word curriculum and the grammatical information that will be required for words of these word classes natural language processing application for that language. After significant of word classes for English and the grammatical information that is required from the words of these word classes, various paradigms for these word classes were developed. Paradigm for a root word gives information about its achievable word forms in a particular word class, and their relevant grammatical information. All the words of a word class may not follow the same paradigm, like; it is not that all nouns will follow the same inflectional pattern. So, the first assignment was to find out the various paradigms for a word class and then group the words of that word class according to those paradigms. Proceeding this way paradigms were developed for the word classes which show inflection. For developing the paradigms the inflectional patterns of the root words of a word class were studied. And, then on their basis, the root words which inflect in the similar way were grouped. The inflection patterns for those groups constitute the set of paradigms for that word classes. Following is the list of word classes along with their grammatical information that are being used for English.

Noun	Grammatical information required for English is –Number, gender, type, and syntactic features. Nouns have singular and plural forms. Many plural forms have -s or -es endings (dog/dogs, referee/referees), in English, nouns do not have grammatical gender. However, many nouns can refer to masculine or feminine animate objects (mother/father, tiger/tigress, male/female). Nouns have several syntactic features that can aid in their identification. The natural language English has noun which indicates the name of the persons, things, etc.	Nouns (example: common noun "cat") may be modified by adjectives ("the beautiful Angora cat"), preceded by determiners ("the beautiful Angora cat"), or pre-modified by other nouns ("the beautiful Angora cat").
Verb	Verb form the second largest word class after nouns. According to Carter and McCarthy, verbs denote "actions, events, processes, and states." Consequently, "smile," "stab," "climb," "confront," "liquefy," "wake," "reflect" are all verbs. verb is used to describe the action or activity of noun.	Some examples of verb endings, which while not dead giveaways, are often associated, include: "-ate" ("formulate"), "-iate" ("inebriate"), "-ify" ("electrify"), and "-ize" ("sermonize"). There are exceptions, of course: "chocolate" is a noun, "immediate" is an adjective, "prize" can be a noun, and "maize" is a noun. Prefixes can also be used to create new verbs. Examples are: "un-" ("unmask"), "out-" ("outlast"), "over-" ("overtake"), and "under-" ("undervalue"). Just as nouns can be formed from verbs by conversion, the reverse is also possible.
Adjectives	Adjectives describe properties, qualities, and states attributed to a noun or a pronoun. As was the case with nouns and verbs, the class of adjectives cannot be identified by the forms of its constituents. However, adjectives are commonly formed by adding the some suffixes to nouns.	Examples: "-al" ("habitual," "multidimensional," "visceral"), "-ful" ("blissful," "pitiful," "woeful"), "-ic" ("atomic," "gigantic," "pedantic"), "-ish" ("impish," "peckish," "youngish"), "-ous" ("fabulous," "hazardous"). Adjectives can also be formed from other adjectives through the addition of a suffix or more commonly a prefix: weakish, implacable, disloyal, irredeemable, and unforeseen. A number of adjectives are formed by adding "a" as a prefix to a verb: "adrift," "astride," "awry."
Adverb	Adverbs are a class of words "which perform a wide range of functions. Adverbs are especially important for indicating time, manner, place, degree, and frequency of an event, action, or process." They typically modify verbs, adjectives, or other adverbs. Adjectives and adverbs are often derived from the same word. A majority of adverbs are formed by adding to "-ly" ending to their corresponding adjective form. Recall the adjectives, "habitual", "pitiful", "impish".	Some suffixes that are commonly found in adverbs are "-ward(s)" and "-wise": "homeward": "The ploughman homeward plods his weary way." "downward": "In tumbling turning, clustering loops, straight downward falling, ..." "lengthwise": 2 to 3 medium carrots, peeled, halved lengthwise, and cut into 1-inch pieces.

Table 1. Delineate of Grammatical segment

Stemmers are used to convert inflected words into their root or stem. Stem does not necessarily correspond to linguistic root of a word. Stemming improve performance by reducing morphologically variants into same words. There are few rules when using medical roots, "o" always acts as a joint-stem to connect two consonantal roots, e.g. *arthr+o+logy= arthrology.* But generally, the "o" is dropped when connecting to a vowel stem, e.g. *arthr+itis=arthritis, instead of arthr-o-itis.*
The list of some roots, suffixes and prefixes used in medical terminology are shown below in table 1.

Words	Prefix	Suffix	Stem/Root Words
Treatment	------	-ment	Treat
Illness	------	-ness	Ill
Stitching	St	-ing	Itch
Hypogastria	Hypo	-tria	Gas
Abortion	------	-tion	Abort
Abscesses	------	-es	Abscess
Hypertension	Hyper	-sion	Tense

Table 1. Root words of Clinical Terminology.

Rules for Suffix
There are certain rules for suffix of the words ending with 'able','ment','ing',etc… the rules are as follows

1. Rules for suffix 'able' are as follows

a) If in a word before 'able', 'b' comes with vowel 'i' then replace 'able' by 'e'
Example, describable ⟶ descr**ib**+able ⟶ describe
ascribable ⟶ ascr**ib**+ able ⟶ ascribe

b) If in a word before 'able', 'b' comes with any consonant or vowel (expet 'b') then remove 'able'.
Example, absorbable ⟶ abso**rb**+able ⟶ absorb
climbable ⟶ cli**mb**+ able ⟶ climb

c) If in a word before 'able', 'h' comes with any consonant or vowel then remove 'able'
Example, abolishable ⟶ abolis**h**+able ⟶ abolish
accomplishable ⟶ accomplis**h**+able ⟶ accomplish

2. Rules for suffix 'ment' are as follows

a) If in a word suffix'ment' comes then remove 'ment'.
Example, abandonment ⟶ abandon+ment ⟶ abandon
establishment ⟶ establish+ment ⟶ establish

3. Rules for suffix 'ly' are as follows

a) If in a word suffix 'ly' comes then remove 'ly'.
Example, kindly ⟶ kind+ly ⟶ kind
softly ⟶ soft+ly ⟶ soft

4. Rules for suffix 'ness' are as follows

a) If in a word suffix 'ness' comes then remove 'ness'.
Example, cleverness ⟶ clever+ ness ⟶ clever
darkness ⟶ dark+ness ⟶ dark

Rules for Prefix
There are certain prefixes such as dis, im, in, mis, pre, re, un,…etc rules for prefix is shown below

a) If in a word prefix 'dis' comes then remove 'dis' from the word

Examples, disagree ⟶ dis+agree ⟶ agree
disorder ⟶ dis+order ⟶ order

b) If in a word prefix 'im' comes then remove 'im' from the word.
Example, impatient ⟶ im+patient ⟶ patient
Impossible ⟶ im+possible ⟶ possible

Existing work on stemmer

Documents are generally represented in terms of the words they contain, as in the vector-space model [2]. Many of these words are similar to each other in the sense that they denote the same concept(s), i.e., they are semantically similar. Generally, morphologically similar words have similar semantic interpretations, although there are several exceptions to this, and may be considered equivalent. The construction of such equivalence classes is known as stemming. A number of stemming algorithms or stemmers, which attempt to reduce a word to its stem or root form, have been developed. Thus, the document may now be represented by the stems rather than by the original words. As the variants of a term are now conflated to a single representative form, it also reduces the dictionary size, which is the number of distinct terms needed for representing a set of documents. A smaller dictionary size results in savings in storage space and processing time.

Stemming is often used in information retrieval because of the various advantages it provides [3]. The literature is divided on this aspect, with some authors finding stemming helpful for retrieval tasks [3], while others did not find any advantage [4]. However, they are all unanimous regarding the other advantages of stemming. Not only is the storage space for the corpus and retrieval times reduced but recall is also increased without much loss of precision. Moreover, the system has the option for query expansion to help a user refine his/her query.

Different Stemming Algorithms

Various stemmers are available for several languages, including English. The most prominent ones are those introduced by Lovins, Dawson, Porter, Krovetz, Paice/Husk and Xu, and Croft. We now provide a brief description of some of these algorithms.

1. Truncate(n): This is a trivial stemmer that stems any word to the first n letters. It is also referred to as n-gram stemmer [5]. This is a very strong stemmer. However, when n is small, e.g., one or two, the number of overstemming errors is huge. For this reason, it is mainly of academic interest only. In this paper, we have chosen n to be 3, 4, and 5 and refer to them as trunc3, trunc4 and trunc5, respectively.
2. Lovins Stemmer: The Lovins stemmer [6] was developed by Lovins and is a single-pass longest match stemmer. It performs a lookup on a table of 294 endings, which have been arranged on a longest match principle. The Lovins stemmer removes the longest suffix from a word. Once the ending is removed, the word is recoded using a different table that makes various adjustments to convert these stems into valid words. However, it is highly unreliable and frequently fails to form words from the stems or to match the stems of like-meaning words.
3. Dawson Stemmer: The Dawson stemmer [7], which was developed by Dawson, extends the Lovins stemmer. This is also a single-pass longest match algorithm, but it uses a much more comprehensive list of around 1200 suffixes, which were organized as a set of branched character trees for rapid access. In this case, there is no recoding stage, which had been found to be unreliable.
4. Porter Stemmer: Porter proposed the Porter stemmer [8], which is based on the idea that the suffixes in the English language (approximately 1200) are mostly made up of a combination of smaller and simpler suffixes. It has five steps, and within each step, rules are applied until one of them passes the conditions. If a rule is accepted, the suffix is removed accordingly, and the next step is performed. The resultant stem at the end of the fifth step is returned.
5. Paice/Husk Stemmer: The Paice/Husk stemmer [9] is a simple iterative stemmer and uses just one table of rules; each rule may specify either deletion or replacement of an ending. The rules are grouped

into sections that correspond to the final letter of the suffix, making the access to the rule table quicker. Within each section, the order of the rules is significant. Some rules are restricted to words from which no ending has yet been removed. After a rule has been applied, processing may be allowed to continue iteratively or may be terminated.

6. Krovetz Stemmer: The Krovetz stemmer [10] was developed by Krovetz and makes use of inflectional linguistic morphology. It effectively and accurately removes inflectional suffixes in three steps: the conversion of a plural to its singular form, the conversion of past to present tense, and the removal of -ing. The conversion process first removes the suffix and then through the process of checking in a dictionary for any recoding, returns the stem to a word. It is a light stemmer in comparison to the Porter and Paice/Husk stemmers.

7. Co-Occurrence-Based Stemmer by Xu and Croft: Xu and Croft [5] observed that most stemmers perform understemming or overstemming, or even both. Strong stemmers generally perform overstemming only. Xu and Croft came up with an algorithm that would refine the stemming performed by a strong stemmer. To this end, they computed the co-occurrences of pairs of words that belong to the same equivalence class. For each pair, they also computed the expected number of co-occurrences, which would account for words that occur together randomly. Thus, they obtained a measure that is similar to the mutual information measure

8. Dictionary-Based Stemmers: There have also been dictionary-based stemmers [3], [11], [12] that improve on an existing stemmer by employing knowledge obtained from a dictionary. Word co-occurrences in a dictionary are considered to imply the relations between words.

9. Probabilistic Stemmers: Given a word in a corpus, the most likely suffix–prefix pair that constitutes the word is computed [13]. Each word is assumed to be made up of a stem (suffix) and a derivation (prefix), and the joint probability of the (stem, derivation) pair is maximized over all possible pairs constituting the word. The suffix and prefix are chosen to be nonempty substrings of the given word, and it is not clear what should be done in the case when a word should be stemmed to itself.

10. Refinement of an Existing Stemmer: In some cases, errors produced by a stemmer are manually rectified by providing an exception list [10]. The stemmer would first look up the exception list, and if the word is found there, it returns the stem found there. Otherwise, it uses the usual stemmer. The aforementioned co-occurrence-based stemmer is also one such algorithm where the exceptions are obtained automatically.

11. Distributional Clustering as Stemming: Distributional clustering [14], [15]–[16] joins (distributionally) similar words into a group if the words have similar probability distributions among the target features that co-occur with them. In the distributions are estimated by observing the grammatical relationships between words and their contexts, whereas, the distributions are obtained from the frequency of words in each category of the corpus. In their work on document classification, Baker and McCallum had chosen the class labels as the target features. The root forms of the words are not taken into consideration while grouping them. This algorithm described is given as follows. The mutual information of each word in the corpus with the class variable is computed, and the words are sorted in descending order. The number of desired clusters is fixed beforehand, e.g., to M. The first M words are initialized to form M singleton clusters. The two most similar (of the M) clusters are merged. This similarity is measured in terms of the Kullback–Leibler divergence of the distributions of the two clusters. The next word in the sorted list forms a new singleton cluster. Thus, the number of clusters remains M each time. In this paper, we refer to Baker and McCallum's method as baker. In our implementation, we have fixed M to the number of stems obtained by refining the trunc3 stemmer using our model.

Features of Proposed Stemmer

Stemmer for clinical language has windows platform. It has unproblematic to use GUI (Graphical User Interface) for the user to operate and need not to have much knowledge about computers, platforms and any programming language. Users just need some essential computer operation knowledge for software installation and manoeuvre. If we confer from the technical point of view, it has been developed using Visual Basic as Front End and Oracle10g as Back End. It is easy to use and give accurate root words. The

proposed stemmer may be useful in medical field which is usually done during document indexing by removing word endings or suffixes using tables of common endings and heuristics about when it is appropriate to eliminate them. Following stature shows the stream of words in database.

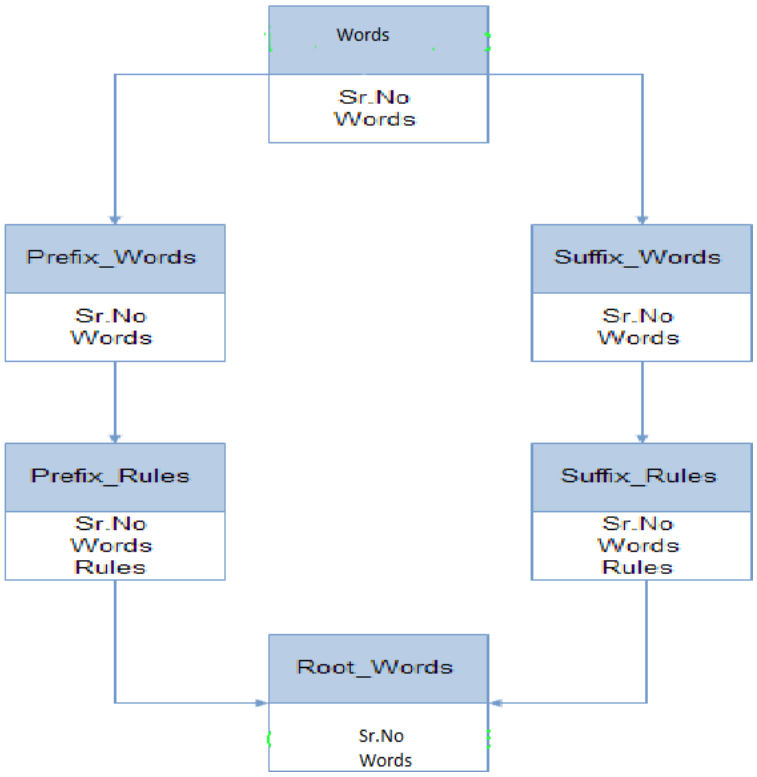


Figure 2. UML diagram of database

We need to develop a new stemmer because the active stemmers which are Algorithm based are not able to give correct root words in some of the words The stemmers are completely based on general languages (regional, communicative), but the clinical terminology is someway diverse from the wide-ranging languages, and the stemmers which are database based are not equipped to give the proper root words of clinical terminology. Consequently we have urbanized the new stemmer based on database which will bestow the appropriate output.

Conclusion

The English clinical stemmer discussed in this paper stores all the commonly used suffix and prefix for all clinical root words in its database. This approach prefers time and accuracy to memory space. We confer some of the rules used to remove suffix and prefix from the clinical words to get the root word. Advantage of this approach is that the user will get the precise results. As sometimes suffix trimming approach in active stemmer provide possible root can result in some extra and indifferent result also. Therefore, this approach is suggested at least for the clinical language in which the number of possible inflections for a word is not infinite.

References

[1] Robert Krovetz. Viewing morphology as an inference process. *In proceedings of the 16th International conference on research and Development in Information Retrieval,* pages 191-202, 1993

[2] G. Salton, A. Wong, and C. S. Yang, "*A vector space model for automatic indexing*", Communications. ACM, vol. 18, no. 11, pp. 613-620, Nov.1975.

[3] W. Kraaij and R. Pohlmann, "*Viewing stemming as recall enhancement*," in Proc. 17th ACM SIGIR Conference., Zurich, Switzerland, Aug. 1996, pp. 40-48.

[4] D.Harman, " *How effective is suffixing?*" J. Amer. Soc. Information Science., Vol. 42, no. 1, pp. 7-15.

[5] J. Xu and W. B. Croft, " *Corpus-based stemming using coocurrence of word variants*", ACM Transactions on Information Systems, vol. 16, no. 1, pp. 61-81,1998.

[6] J. B. Lovins, *"Development of a stemming algorithm,"* Mechanical Translation and Computer Linguistic., vol.11, no.1/2, pp. 22-31, 1968.

[7] J. L. Dawson, *" Suffix removal for word conflation,"* Bulletin of the Association for Literary and Linguistic Computing., vol. 2, no. 3, pp. 33-46, 1974.

[8] M. F. Porter, " *An Algorithm for suffix stripping*," Program, vol. 14, no. 3, pp. 130-137, 1980.

[9] C. D. paice, " *Another stemmer*," SIGIR Forum, vol. 24, no. 3, pp. 56-61, 1990.

[10] R. Krovetz, "*Viewing morphology as an inference process*," in Proceedings. 16th ACM SIGIR Conference., Pittsburgh, PA, 1993, pp. 191-202.

[11] M. Kantrowitz, B. Mohit, and V. Mittal, "*Stemming and its effects on TFIDF ranking*," in Proceedings. 23rd Annual for SIGIR Conference. Athens, Greece, 2000, pp. 357-359.

[12] T. Gustad and G. Bouma, "Accurate *stemming of Dutch for text classification*," Language and Computers., vol. 45, no. 1, pp. 104-117, 2002.

[13] M. Bacchin, N. Ferro, and M. Melucci, "*A probabilistic model for stemmer generation*," Information Processing and Management., vol. 41, no. 1, pp. 121-137,2005.

[14] L. D. Baker and A. K. McCallum, "*Distributional clustering of words for text classification*," in Proceedings. 21st ACM SIGIR Conference., Melbourne, Australia, 1998, pp. 96-103.

[15] F. Pereira, N. Tishby, and L. Lee, "*Distributional clustering of English words*," in Proceedings. 31st Annual meeting on Association for Computational Linguistics. 1993, pp. 183-190.

[16] L. Lee, "*Measures of distributional similarity*," in Proceedings. 37th Annual Meeting on Association for Computational Linguistics. 1999, pp. 25-32.

Lexipedia: A Multilingual Digital Linguistic Database

Rajesha N
Senior Technical Officer,
ldc-rajesha@ciil.stpmy.soft.net

Ramya M
Senior Technical Officer,
ldc-ramya@ciil.stpmy.soft.net

Samar Sinha
Senior Lecturer / Junior Research Officer
ldc-samar@ciil.stpmy.soft.net

Linguistic Data Consortium for Indian Languages
Central Institute of Indian Languages
Mysore, India
www.ldcil.org

***Abstract*: Lexipedia, a multilingual digital linguistic database aims to provide all types and kinds of information that a linguistic item carries in a language, and its cross-linguistic morphemic equivalent in other languages. It provides a wide range of information from graphemic to idiomatic expressions and beyond. In this paper, Lexipedia is conceptualised as a model of human knowledge of language, and its description and architecture is an effort towards modelling such linguistic knowledge.**

I. Lexical Database: issues and limitations

For more than 2000 years, paper dictionaries are compiled with a view to provide specific information that it aims to provide. Hence, there are several types of dictionaries providing specific information depending upon the type of dictionary. Similarly, electronic/digital dictionary does the same by replacing the format. An electronic dictionary, though primarily designed to provide basic information such as grammatical category, meaning, usage, frequency, etc., has also got its usage in various other ancillary tasks in the newer domains of language use. Such electronic dictionary, however, has a major shortcoming as it provides specific information considering the scope, usage, and storage for which it is developed. In other words, other different kinds of information that the language users require are often not featured but are readily available in another dictionary specifically created for it. In another aspect, such dictionary is a mere list of lexical items with its specific information, and does not reflect how human beings store and process such lexical items.

With the advent of newer domains of language use, however, different kinds of resources are conceptualised and designed to store information which serve as database for different kinds of applications and processes. One such electronic lexical database is WordNet, which organises words into sets of cognitively synonymous sets (often called synsets [1] and [2].) It stores lexical items of a language hierarchically and the conceptual-semantic and lexical semantic relationships between these items are determined cognitively. In other words, it is a hybrid of dictionary and thesaurus providing information of the both. However, the major concern for which Princeton cognitive psychologist George A. Miller developed WordNet is to model a database that is consistent with the knowledge acquired about how human beings process language. In addition to it, WordNet is interpreted and used as ontology. Despite its wider use in several applications like Word Sense Disambiguation (WSD), Information Retrieval (IR), automatic text classification, automatic text summarization, etc., WordNet like other lexical databases too has its own limitations.

These databases are designed with certain specific objectives, hence, to access the detailed information about a particular linguistic item one has to access several different kinds of databases specifically meant to provide the required specific information. For example, to access detailed information about a word 'किताब' in Nepali, one has to access WordNet for conceptual-semantic and lexical-semantic relations, pronunciation dictionary, or even separate databases for usage, idioms, proverbial usage, etc. Similarly, if one has to find its equivalent in other languages, one has to scan bi/multilingual dictionary. As it is known, accessing different databases often lead to inconsistency since each database is constructed to fulfill certain objective. Moreover, such databases are primarily not designed to provide different kinds of information that a Natural Language Process system requires. In other words, it is imperative to build a consistent, uniform, dedicated database which serves NLP applications.

In section 2, the paper explores conceptual design and organisation of different fields, which are modularised with respect to specific information. A principled basis of comparing various linguistic phenomenon across languages and to achieve such an objective to avoid miss-comparison, and in creating typological databases are the subject matter of the following section. Section 4 deals with the computational aspect along with the design of the back-end and algorithms to execute various information. One of the input interfaces is also highlighted in building such database. The final section is a summary.

II. Lexipedia: concept and organisation

In view of the above shortcomings of the lexical databases, Lexipedia is conceptualised to provide all and every kind of information that a particular linguistic item in a particular language embeds, and its cross-linguistic morphemic equivalent in other languages. Here, it is imperative to mention that linguistic item includes free forms as well as bound forms. The latter is the result of grammaticalisation, a historical processes resulting various forms, functions and constructions (see [3] and [4]).

Lexipedia is designed to model how humans organise these linguistic items, and in turn how these items are related with each other as well as with its linguistic usage in various other forms, functions and constructions in a language. In other words, it is designed to reflect all kinds of information that a user of a language carries overtly/covertly over the synchronic/diachronic dimension about a particular item in a language, and its morphemic equivalent across languages. Lexipedia, hence, provides wide ranging information on a linguistic item which is organised in modules.

Since, information that Lexipedia provides is wide and vast, it is organised into different modules, where each module provides specific information regarding an item. Having such a modular architecture for information organisation has an advantage as each module can be customised according to the need of the application/users as well as for resource building. These modules are designed as follows:

A. Graphemic

An item's scriptal graphemic information is provided following the script used for a particular language like Devanagari for Hindi, Nepali, Marathi, Bodo, etc.; Srijanga script for Lepcha, etc. It also provides spelling variations if an item has in a particular language. Along with it, transliteration of the item following the LDCIL transliteration scheme and the (broad) IPA transcription are also provided.

B. Audio-video

Audio-video information about a linguistic item is provided at another module. In this module, pronunciation in audio file, and in cases, image/video files are also supplemented. This module is handy in the study of sub-lexical structure of a language as well as for developing pronunciation dictionary, and other speech related applications.

C. Grammatical

Grammatical information forms the basis of various NLP applications. The grammatical categories are noun, pronouns, verb, adjectives, adverbs, adposition, and particles, which subsumes a larger number of other traditionally defined categories like conjunction, interjection, clitics, etc. In Lexipedia, the grammatical information for each category is provided in hierarchical layers. For example, nouns are organised with respect to the categorising device that language employs (gender, classifier, number, honorificity, etc.). To illustrate such a noun categorisation, Hindi and Assamese employ gender and classifier, respectively. Among the Tibeto-Burman languages, Khasi and Lepcha are other two languages which extensively organises nouns on the basis of classifiers. Similarly, verbs are typologised and organised on the basis of their syntactic behavior into types following [5] To cite an example, Hindi verbs can be typologised following [6] In the case of adjectives, the Cinque Hierarchy (see [7]) can be explored for Indian languages.

In addition to this information about the categories, Lexipedia also provides information on different grammatical categories like tense, aspect, mood, aktionsart, case markers, voice, classifier, gender, person, number, clusivity, etc.

D. Semantic

In this module, multiple semantic information is provided for which Lexipedia employs corpora to ascertain meaning both in its synchronic and diachronic dimensions. Such semantic variation is supplemented by the citation of the actual usage from the corpora.

E. Other

Lexipedia also records proverbial, idiomatic, register, domain specific and various other usages of an entry. Hence, it provides information on various uses of the entry in a language also. At the same time, it also provides information on root, lexical stock and etymology of an entry. Similarly, lexical semantic relations are also presented forming ontology of organisation of items in a particular language.

III. Cross-linguistic typology

One of the major decisions regarding providing cross-linguistic information is about the uniformity of phenomenon in question, and to handle various gradient linguistic phenomena in a principled way. Since Lexipedia provides cross-linguistic information across Indian languages, it is imperative to follow a uniform definition of grammatical category across these languages to arrive at true cross-linguistic information on Indian languages. In pursuit of such cross-linguistic uniformity, it is essential to adopt standards that can be applied uniformly across languages and which allow to compare like with like. Moreover, such standard should also ensure that the cross-linguistic study of the phenomenon is not missed out either due to the different labels or we compare different phenomena due to the same label.

In order to achieve such criteria, Canonical approach, which is put forward to account typology of possible words in the realm of typology, and is widely used in the realm of morphology and syntax is best suited. Canonical approach takes definitions to their logical end point and builds theoretical spaces of possibilities, and creates theoretical spaces, to populate them while the languages are still there to be investigated. Moreover, it is also useful to study both what is frequent and what is rare, and in the construction of typological databases.

IV. At the back-end

Since Lexipedia is a multilingual database, and has many-to-many relations across languages, scripts, orthography, fields and entries, it throws an enormous challenge for computational and programming aspects. To accomplish

such linkages, we have basically adopted a model which is based on concept related to the linguistic item. In this model, concept refers to a description of an item in a link language. For our present purpose, owing to pragmatic factors, we have identified it to be English. To cite an example, a linguistic item in Kannada 'kEsarI' (ಕೇಸರಿ) has three set of concepts.

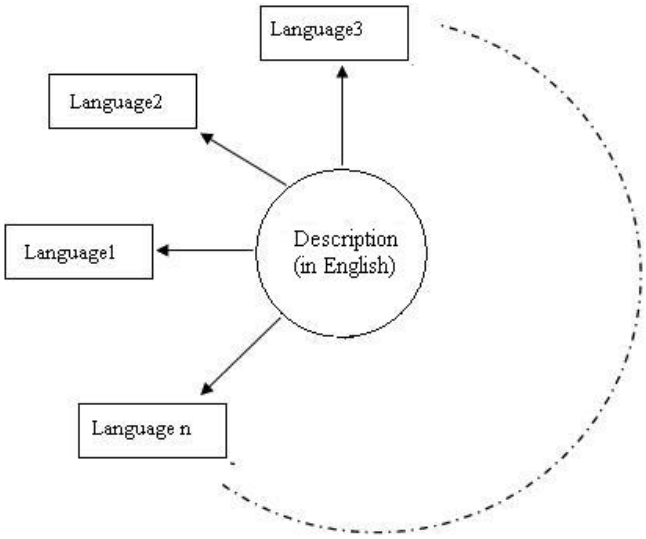
A shade of yellow tinged with orange (SAFFRON).

A flavoring agent (SAFFRON).

A large tawny flesh-eating wild cat of Africa and South Asia (LION).

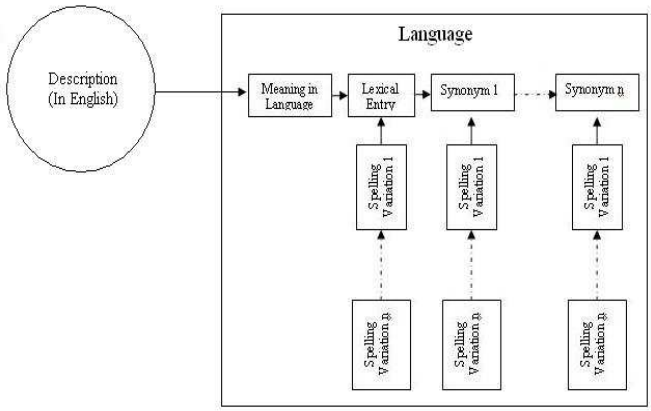
In Lexipedia, rather than following the equivalent items across languages, the descriptive meaning of the item in question is followed. In other words, based on equivalent meaning, items are interrelated, and iterated over different languages. Under such approach, however, it is a known fact that lexical under-specification across languages is encountered. To account such issue, the descriptive meaning of the item in the question will be considered for providing linkages across languages.

Based on the 'descriptive meaning (in English)', the process is iterated in other languages. In other words, we are following indexation of 'descriptive meaning (in English)'.



Conceptual Representationof Linking Languages

In Lexipedia, we have adopted a 'description set model' i.e. based on description (descriptive meaning in English), we provide the entry, meaning (in the language), spelling variation of the entry, and synonyms of the entry. In other words description set consists of description in English, its spelling variations, and synonyms and their respective spelling variations, and meaning in the language where all these items share among each other.



Graphical Representation of Description Set

Other lexical semantic relations are entered manually. IPA, pronunciation, and transliteration (following the LDCIL scheme v0.1) are embedded in the system. To expedite the data entry, we have developed graphical user interface (GUI) which automatically picks 'description set model's' synonyms and spelling variations as an entry and other fields are provided manually.

For the management of Lexipeida, we have devised a methodology that only one language should add fresh concepts (Description in English) at a given point of time. Such language will be called as Primary Language (PL). All other languages will add the entries and other respective fields in their language in correspondence with the concepts given by the PL. We have developed two text data input interfaces for Lexipedia [snapshots are in Annexure I] for both PL and Secondary Language (SL) entry.

V. SUMMARY

Lexipedia attempts to provide wide ranging information, and caters the needs of a user about a specific linguistic item in a language, and its morphemic equivalent across languages. Unlike other lexical databases, it provides information at different levels from graphemic to idiomatic expressions and beyond. Its architecture is modular; hence, it can be customised according to the needs of the specific applications/users.

In its conceptualisation and design, Lexipedia provides specific information of an item at the strata called levels that can be customised according to the requirements. Each level provides specific information.

Lexipedia serves as a linguistic resource hub for Indian languages (at this level of development), however, it can be enriched with other languages, drawing cross-linguistic morphemic similarities and differences between languages. On the other hand, it is conceptualised as a model of what a native speaker of a language knows about an item in his/her language synchronically/diachronically. Lexipedia is an effort towards modeling such linguistic knowledge.

ACKNOWLEDGMENT

We would like to thank Dr. B. Mallikarjun, who initially floated the idea of creating multilingual dictionary of Indian languages - a precursor to Lexipedia, and contributed valuable inputs into Lexipedia. We are grateful to Prof. Kavi Narayana Murthy (CIS, UoH, Hyderabad; currently CIIL fellow) for his guidance, help, insightful comments and suggestions on the different issues. We are heartily thankful to our Project Head, Dr. L. Ramamoorthy, whose encouragement, guidance and support enabled us to sum up our efforts so far into words, and other members of the Team LDC-IL for their comments and relevant help.

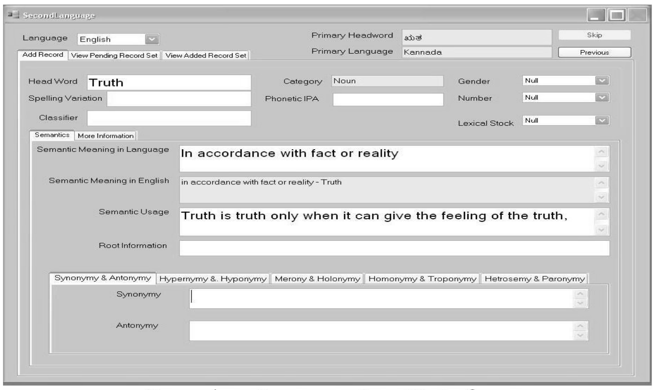
REFERENCES

[1]. Miller, George A. 1995. WordNet: A Lexical Database for English. *Communications of the ACM Vol. 38, No. 11*. pp. 39-41.
[2]. Fellbaum, Christiane . 1998.*WordNet: An Electronic Lexical Database.*The MIT Press.
[3]. Lehmann, Christian. 1995. *Thoughts on Grammaticalization*. Munich: Lincom Europa.
[4]. Hopper, Paul J., and Elizabeth Closs Traugott. 1993.*Grammaticalization*. Cambridge, England: Cambridge University Press.
[5] Levin & Rappaport. 1995. *Unaccusativity: At the Syntax-Lexical Semantics Interface*. Linguistic Inquiry Monograph 26, MIT Press, Cambridge, MA.
[6]. Richa. 2008. Unaccusativity, Unergativity and the Causative Alternation in Hindi: A Minimalist Analysis. Ph.D thesis, Jawaharlal Nehru University, New Delhi.
[7]. Cinque, Guglielmo. 1999. Adverbs and Functional Heads. Oxford: OUP

Annexure I

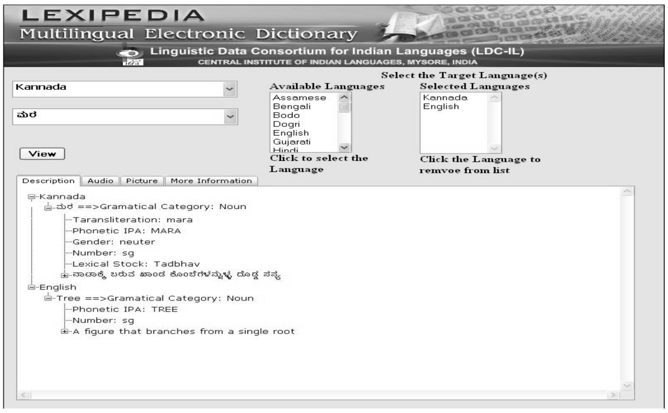
Lexipedia Input Interfaces

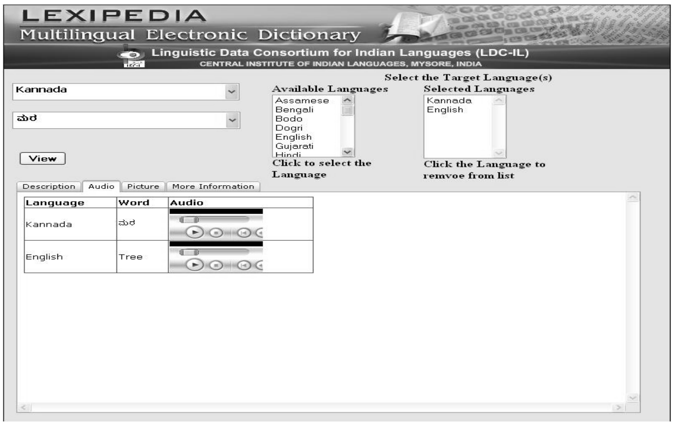
Primary Language Input Interface

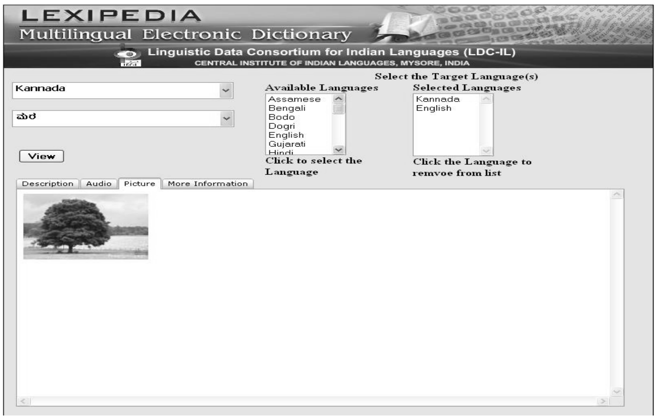


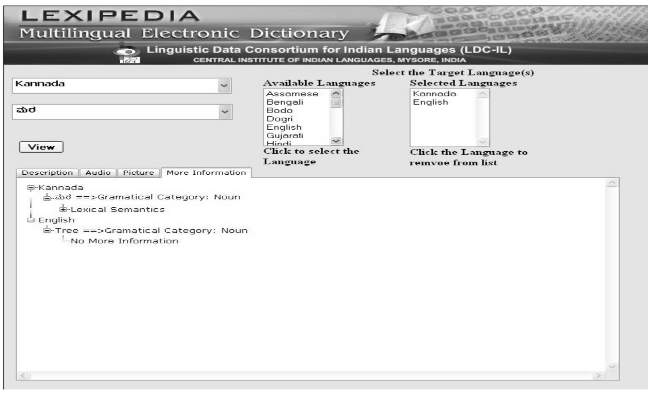
Secondary Language Input Interface

Output Interfaces developed in First Version.









Text Extraction for an Agglutinative Language

Sankar K, Vijay Sundar Ram R and Sobha Lalitha Devi
AU-KBC Research Centre
MIT Campus of Anna University
Chennai, India

***Abstract*- The paper proposes an efficient algorithm for sentence ranking based on a graph theoretic ranking model applied to text summarization task. Our approach employs word frequency statistics and a word positional and string pattern based weight calculation for weighing the sentence and to rank the sentences. Here we have worked for a highly agglutinative and morphologically rich language, Tamil.**

I. INTRODUCTION

The enormous and on-going increase of digital data in internet, pressurize the NLP community to come up with a highly efficient automated text summarization tools. The research on text summarization is boosted by the various shared tasks such as TIPSTER SUMMAC Text Summarization Evaluation task, Document Understanding conference (DUC 2001 to 2007) and Text Analysis conferences.

A variety of automated summarization schemes have been proposed recently. NeATS [4] is a sentence position, term frequency, topic signature and term clustering based approach and MEAD [10] is a centroid based approach. Iterative graph based Ranking algorithms, such as Kleinberg's HITS algorithm [3] and Google's PageRank [1] have been successfully used in web-link analysis, social networks and more recently in text processing applications [8], [7], [2] and [9]. These iterative approaches have a high time complexity and are practically slow in dynamic summarization. The works done in Text Extraction for Indian languages is comparatively less.

In this paper we have discussed a novel automatic and unsupervised graph based ranking algorithm, which gives improved results compared to other ranking algorithms in the context of the text summarization task. Here we have worked for Tamil.

II. TEXT SUMMARIZATION AND TEXT RANKING

Text summarization is process of distilling the most important information from the set of source to provide a abridge version for particular user and tasks. The text summarization is also done by ranking in the sentences in the given source test. Here we have proposed a graph based text ranking approach.

Graph based algorithm is essentially a way of deciding the importance of a vertex within a graph, based on global information recursively drawn from the entire graph. The basic idea here is that of 'voting' or 'recommendation'. When one vertex links to the other vertex, it is like casting a vote for that vertex. The vertex becomes important when it links with more number of vertices. The importance of vertex casting the vote determines how important the vote itself is [10].

The proposed graph based text ranking algorithm consists of two types of measure (1) Word Frequency Analysis; (2) A word positional and string pattern based weight calculation. Based on the above two scores, the ranking of sentences is done.

The algorithm is carried out in two phases. The weight metric obtained at the end of each phase is averaged to obtain the final weight metric. Sentences are sorted in descending order of weight.

A. Graph

Let G (*V*, *E*) be a weighted undirected complete graph, where *V* is set of vertices and *E* is set of weighted edges.

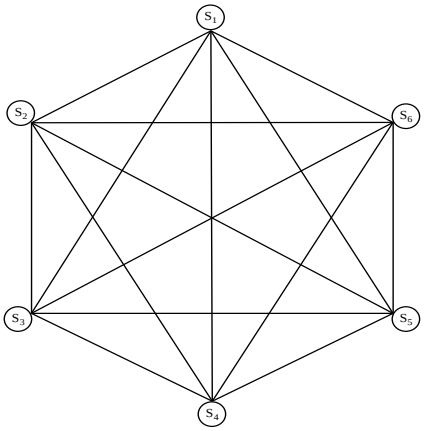


Fig. 1. A complete undirected graph

In figure 1, the vertices in graph G represent the set of all sentences in the given document. Each sentence in G is related to every other sentence through the set of weighted edges in the complete graph.

B. Phase 1 : Word Frequency Analysis

In Word Frequency Analysis, we find the affinity weight (AW) for each word in the sentence by using the formula 1. The sentence weight (SW) is calculated by averaging the AW of all words in the sentence.

The affinity weight for each word is calculated by frequency of the given word in the sentence divided by number of words in the sentence.

Word Frequency in Tamil:

As Tamil is a morphologically rich and a highly agglutinative language, getting the frequency of the words is not straight forward. The text has to be preprocessed with a morph-analyser to collect the corresponding root words, as all the words in the sentences will be in inflected form (root + suffixes). Given a word to the morph-analyser, it will split the word into root and its suffix and return the valid root word alone. Example

மரங்களை -> மரம் + கள் + ஐ -> மரம்

marangkaLai -> maram + kaL + ai -> maram

(tree + plural +acc) tree plural acc tree

Let the set of all sentences in document $S = \{s_i \mid 1 \le i \le n\}$, where n is the number of sentences in S. For a sentence $s_i = \{w_j \mid 1 \le j \le m_i\}$ where m_i is the number of words in sentence s_i, $(1 \le i \le n)$ the affinity weight AW of a word w_j is calculated as follows:

$$AW(w_j) = \frac{\sum_{\forall w_k \in S} IsEqual(w_j, w_k)}{WC(S)} \qquad (1)$$

where S is the set of all sentences in the given document, w_k is a word in S, WC (S) is the total number of words in S and function $IsEqual(x, y)$ returns an integer count 1 if x and y are equal else integer count 0 is returned by the function.

Then, we find the sentence weight SW (s_i) for each sentence s_i $(1 \le i \le n)$ as follows:

$$SW(s_i) = \frac{1}{m_i} \sum_{\forall w_j \in s_i} AW(w_j) \qquad (2)$$

At the end of phase 1, the graph vertices hold the sentence weight as shown in figure 3 for graph constructed using the following sentences.

[1] தாஜ் மகால், இந்தியாவிலுள்ள நினைவுச்சின்னங்களுள், உலக அளவில் பலருக்குத் தெரிந்த ஒன்றாகும்.

Taj Mahal, among the memorials in India, is known word wide.

[2] இது ஆக்ராவில் அமைந்துள்ளது.

This is located in Agra.

[3] முழுவதும் பளிங்குக் கற்களாலான இக்கட்டிடம், ஆக்ரா நகரில் யமுனை ஆற்றின் கரையில் கட்டப்பட்டுள்ளது.

This building fully made of marbles is built on the shores Yamuna river in Agra.

[4] இது காதலின் சின்னமாக உலகப் புகழ் பெற்றது.

This is world famous as a symbol of love.

[5] ஏழு உலக அதிசயங்களின் புதிய பட்டியலில் தாஜ் மகாலும் சேர்க்கப்பட்டுள்ளது.

In the new seven wonders of the world Taj Mahal is also included.

[6] இக்கட்டிடம் முகலாய மன்னனான ஷாஜகானால், இறந்து போன அவனது இளம் மனைவி மும்தாஜ் நினைவாக 22,000 பணியாட்களைக் கொண்டு 1631 முதல் 1654 ஆம் ஆண்டுக்கு இடையில் கட்டிமுடிக்கப்பட்டது.

Mughal emperor Sharjahan built this building using 22,000 workers, from 1631 to mid of 1654, in memory of young wife Mumthaz .

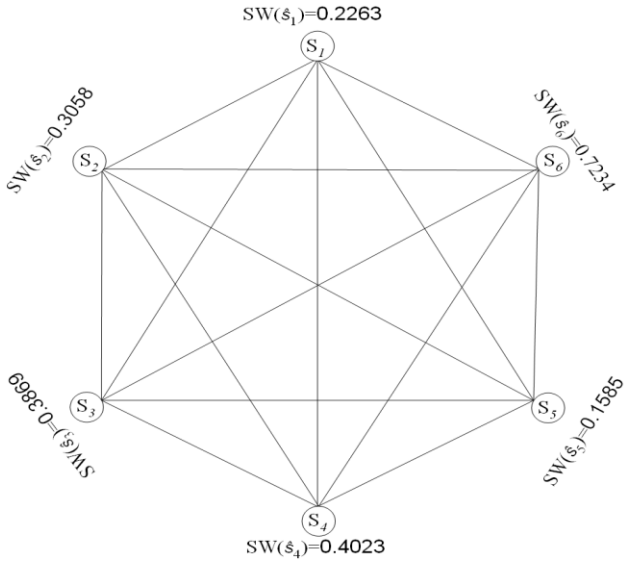


Fig. 3. Sample graph of Sentence weight calculation in phase 1.

C. Phase 2 : A Word Positional and String Pattern Based Weight Calculation

In phase 2, a word positional and string pattern based weight in all the vertices is calculated using Levenshtein Similarity measure (LSW), which uses Levenshtein Distance for calculating the weight.

The vertex weight is calculated by summing all the LSW and dividing it with number of sentences.

Levenshtein Distance

Levenshtein distance (LD) is a measure of the similarity between two strings source (s) and target (t). The distance is the minimum number of deletions, insertions, or substitutions required to transform s into t.

The *LD* algorithm is illustrated by the following example

LD (RAIL, MAIL) is 1

LD (WATER,METER) is 2

Similarly, the LD calculation is same for words in Tamil, but there are three and two character letters in Tamil which we have to consider as single character while calculating the distance, as shown below.

LD(சொல்,வால்) is 1

LD(வரும்படி,என்னப்படி) is 4

Levenshtein Similarity Weight

Levenshtein Similarity Weight is calculated between the sentences, considering two sentences at an instance. This is calculated by dividing the difference of maximum length between two sentences and LD between the two sentences by maximum length between two sentences as shown in formula 6.

Consider two sentences, *sentence*1 and *sentence*2 where ls_1 is the length of *sentence*1 and ls_2 be the length of *sentence*2. Compute *MaxLen*=maximum (ls_1, ls_2). Then *LSW* between *sentence*1 and *sentence*2 is the difference between *MaxLen* and *LD*, divided by *MaxLen*. Clearly, *LSW* lies in the interval 0 to 1. In case of a perfect match between two words, its *LSW* is 1 and in case of a total mismatch, its *LSW* is 0. In all other cases, 0 < *LSW* <1. The *LSW* metric is illustrated by the following example.

Considering these strings as sentences,

LSW (ABC, ABC) =1

LSW (ABC, XYZ) =0

LSW (ABCD, EFD) =0.25

Similarly

LSW (என்னப்படி, வருப்படி) = (6-4)/6 = 0.3334

Levenshtein similarity weight is calculated by the equation

$$LSW(s_i, s_j) = \frac{MaxLen(s_i, s_j) - LD(s_i, s_j)}{MaxLen(\hat{s}_i, s_j)} \qquad (6)$$

where, s_i and s_j are the sentences.

Hence before finding the LSW, we have to calculate the LD between each sentence.

Let $S = \{s_i \mid 1 \le i \le n\}$ be the set of all sentences in the given document; where n is the number of sentences in S. Further, $s_i = \{w_j \mid 1 \le j \le m\}$, where m is the number of words in sentence s_i.

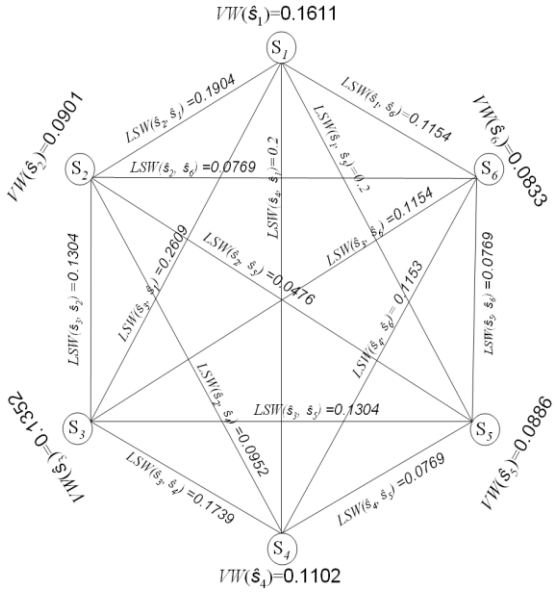


Fig. 4. Sample graph for Sentence weight calculation in phase 2

Each sentence s_i ; $1 \le i \le n$ is represented as the vertex of the complete graph as in figure 4 and $S=\{s_i|1 \le i \le n\}$. For the graph in figure 4, find the Levenshtein similarity weight *LSW* between every vertex using equation 6. Find vertex weight (*VW*) for each string s_i ; $1 \le l \le n$ by

$$VW(s_l) = \frac{1}{n} \sum_{\forall s_i \ne s_l \in S} LSW(s_l, s_i) \qquad (7)$$

3. Text Ranking

Obtaining the sentence weight ($SW(s_i)$) and the vertex weight $VW(s_i)$, the ranking score is calculated is the formula 8, where the average of the two scores are found.

The rank of sentence s_i; $1 \le i \le n$ is computed as

$$Rank(s_i) = \frac{SW(s_i) + VW(\hat{s}_i)}{2}; 1 \le i \le n \qquad (8)$$

where, $SW(s_i)$ is calculated by equation 2 of phase 1 and $VW(\hat{s}_i)$ is found using equation 7 of phase 2. The ranking scores for the sentences (s_i; $1 \le i \le n$,) are arranged in descending order of their ranks.

$SW(s_i)$ in phase 1 holds the sentence affinity in terms of word frequency and is used to determine the significance of the sentence in the overall raking scheme. $VW(\hat{s}_i)$ in phase 2 helps in the overall ranking by determining largest common subsequences and other smaller subsequences then assigning weights to it using *LSW*. Further, since named entities are represented as strings, repeated occurrences are weighed efficiently by *LSW*, thereby giving it a relevant ranking position.

4. Evaluation And Discussion

We have used the ROUGE evaluation toolkit to evaluate the proposed algorithm. ROUGE, an automated summarization evaluation package based on N-gram statistics, is found to be highly correlated with human evaluations [4].

The evaluations are reported in ROUGE-1 metrics, which seeks unigram matches between the generated and the reference summaries. The ROUGE-1 metric is found to have high correlation with human judgments at a 95% confidence level, so this is used for evaluation. The present Graph-based Ranking Algorithms for Text Extraction works with Rouge score of 0.4723.

We manually created the reference summaries for 150 documents taken from online news articles. The reference summaries and the summaries obtained by our algorithm are compared using the ROUGE evaluation toolkit, which is presented in Table 1. For each article, our proposed algorithm generates a 100-words summary.

TABLE I
Rouge Score

	Score
ROUGE-1	0.4723

The methodology performs well even for the agglutinative languages. For the word frequency calculation we feed only the root words instead of the agglutinative words to get proper frequency count. In the phase 2 where the Levenshtein Similarity Weight, the distance varies more as the all the sentences have different inflected and agglutinative words. Again in word frequency, the pronouns occurring in the same sentence, which actual reference to one of the noun phrase (occurs instead of a noun phrase), cannot to be counted.

Conclusions

In this paper, we introduced Graph Based Ranking algorithm for text ranking. Here we have worked for Tamil, a south Dravidian language. Here we have shown the necessity of getting the root words for Text ranking. The architecture of the algorithm helps the ranking process to be done in a time efficient way. This text ranking algorithm is not a domain specific and also does not require any annotated corpora. This approach succeeds in grabbing the most important sentences based on the information exclusively from the text itself; whereas other supervised ranking systems do this process by training on summary collection.

5. CONCLUSIONS

In this paper, we introduced Graph Based Ranking algorithm for text ranking. Here we have worked for Tamil, a south Dravidian language. Here we have shown the necessity of getting the root words for Text ranking. The architecture of the algorithm helps the ranking process to be done in a time efficient way. This text ranking algorithm is not a domain specific and also does not require any annotated corpora. This approach succeeds in grabbing the most important sentences based on the information exclusively from the text itself; whereas other supervised ranking systems do this process by training on summary collection.

REFERENCES

[1] Brin and L. Page. 1998. The anatomy of a large-scale hypertextualWeb search engine. *Computer Networks and ISDN Systems*, 30 (1 – 7).

[2] Erkan and D. Radev. 2004. Lexpagerank: Prestige in multi-document text summarization. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, July.

[3] Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604-632.

[4] Lin and E.H. Hovy. From Single to Multi-document Summarization: A Prototype System and its Evaluation. *In Proceedings of ACL-2002*.

[5] Lin and E.H. Hovy. 2003a. Automatic evaluation of summaries using n-gram co-occurrence statistics. *In Proceedings of Human Language Technology Conference (HLT-NAACL 2003)*, Edmonton, Canada, May.

[6] Lin and E.H. Hovy. 2003b. The potential and limitations of sentence extraction for summarization. *In Proceedings of the HLT/NAACL Workshop on Automatic Summarization*, Edmonton, Canada, May.

[7] Mihalcea. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization. *In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004) (companion volume)*, Barcelona, Spain.

[8] Mihalcea and P. Tarau. 2004. TextRank - bringing order into texts. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, Barcelona, Spain.

[9] Mihalcea, P. Tarau, and E. Figa. 2004. PageRank on semantic networks, with application to word sense disambiguation. *In Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland.

[10] Radev, H. Y. Jing, M. Stys and D. Tam. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40: 919-938, 2004.

Semantic Representation of Causality

Sobha Lalitha Devi
AU-KBC Research Centre
MIT Campus of Anna University
Chennai
sobha@au-kbc.org

Menaka S
AU-KBC Research Centre
MIT Campus of Anna University
Chennai
menakas@au-kbc.org

***Abstract*—This is an attempt to study the semantic relation of Causality or Cause-Effect, how it is marked in Tamil, how the causal markers in Tamil manifest in texts, their syntactic and semantic properties and how this information can be represented so as to handle causal information and reasoning.**

Keywords- causality; Tami; semantic relation; cause-effect.

I. INTRODUCTION

Causality or Cause-Effect relation is a complex semantic relation. It refers to the relation between two events. If an event E2 is caused by another event E1, then a causative relation exists between event E1 and event E2. E1 is the cause of E2 and E2 is the consequence of E1.

I bought a new pen because I lost the old one. - (1)

Here the event E1 is "I lost the old one" and the event E2 is "I bought a new pen". The causality marker "because" connects the two events E1 and E2, thus establishing a Cause-Effect relationship between the two events.

I bought a new pen after I lost the old one. - (2)

In example (2), the events remain the same. But the marker "after" simply specifies a temporal relationship between the two events. Here, there is no Cause-Effect relation. Also, it may be noted that the relationship is asymmetric, i.e, E1 causes E2 does not imply E2 causes E1.

An attempt has been made to study this Cause-Effect relation in Tamil and the various markers which serve to express this semantic relation. The Attribute Value Matrix (henceforth AVM) representations for some of the markers have been drawn for some examples of expressions of cause. This has given an insight into the causal markers in Tamil.

II. PREVIOUS WORK

Several philosophers have studied the semantic relation of causality like (Ehring 1997), Mellor (1995), Owens (1992) and Sosa and Tooley (1993). Though extensive work has been done in the analysis of causality in English, there has not been much work done on causality in Tamil.

From a natural language understanding(NLP) perspective, Khoo (1995) analyzed the verb entries in the Longman Dictionary of Contemporary English (1987) and came up with a total of 2082 causative verbs (verbs with a causal component in their meaning). In subsequent works (Khoo et al, 2001, Khoo and Myaeng, 2002 and Khoo et al 2002), attempts at automatic recognition of cause-effect relations have been made for information retrieval purposes. Nazarenko-Perrin (1993) has attempted to represent causality using conceptual graphs. Girju (2003) has also attempted the automatic recognition of causal relation in English texts. But, there has not been any attempt to study causal relation in Tamil, especially from a computational perspective.

III. ANALYSIS OF CAUSALITY IN TAMIL

Causality or the semantic relation of Cause-Effect in Tamil is expressed in many ways. It can be syntactic (a suffix) or lexical. It can be within a clause, inter-clausal or inter-sentential. The various causal markers and their features are studied and discussed below.

A. Arguments of the Causal marker

The semantic relation of cause holds between two arguments – the cause or the reason and the effect. Consider the following example.

He died of drowning. - (3)
He drowned due to heavy flood. - (4)
He died due to heavy flood. - (5)

In example (3), it may be noted that the Cause is "drowning" and the Effect is "he died".

In example (4), the Cause is "heavy flood" and the Effect is "he drowned".

In example (5), the Cause is "heavy flood", but the effect is "he died".

In other words, "he died because of drowning due to heavy flood". Here we see that the Cause-Effect event chain. Hence we see that for a particular result or effect, we have two causes – a direct Cause and an indirect Cause. Similarly, for a particular cause, we have two effects – an intermediate Effect and an ultimate Effect.

In the above examples, "heavy flood" is the indirect Cause and "he drowned" is the intermediate effect and the direct Cause. The ultimate Effect is "he died".

B. The Markers of Causality

The causal markers in Tamil may be divided into two categories – those markers which have a noun as their Cause argument and those which take a verb as their Cause Arguments.

1) -aal

The predominantly used marker of cause is *-aal*. When it takes a noun as the cause, it manifests as below.

avar maaraTaipp-aal kaalamaanaar.
he heartattack-CAUSE expired.
"He died of heart attack."

But this marker is polysemous. Sometimes, it denotes instrumentality, as below.

avan katti-y-aal kutt-in-aan.
he knife--INS stab-PST-3SM
"He stabbed with a knife."

This marker may add to the verbal stems in the past or future tense to denote cause. But, it is to be noted that the verbal stem is first nominalized with *atu* and then this marker is added with or without the euphonic markers *an* or *in*.

kaaRRu aTi-tt-a-at-aal mazai pey-t-atu.
Wind blow-PST-RP-3SN-CAUSE rain rain-PST-3SN
"It rained because of the wind."

takka neerattil maruttuvamanai-kku ce-nR-a-at-an-aal avar uyir pizai-tt-aar.
correct time hospital-DAT go-PST-RP-3SN--CAUSE he life save-PST-3SH
"His life was saved because he went to the hospital at the right time."

It may be noted that the marker *-aal* attaches to verbal roots without nominalization to form the conditional form and this is different from Causality.

avan paTi-tt-aal veRRi peRu-v-aan.
he study—COND success get-FUT-3SM
"If he studies, he will succeed".

2) kaaraNattaal

This marker literally means "because of the reason". We may note that the causal marker -aal is present in this marker.

avan paTikk-aat-a kaaraNattaal tooRRaan.
He study-NEG-RP CAUSE fail-PST-3SM.
"He failed because he did not study."

3) kaaraNamaaka

This marker also literally means "because of the reason". *kaaraNam* means reason.

iRaiccal-in kaaraNamaaka enakku onRumee keeTkavillai.
Noise-GEN CAUSE I-DAT anything hear-INF-NEG
"I cannot hear anything because of the noise".

4) kaaraNam

This marker means "reason". The peculiarity of this marker is that this is the only marker where the Cause follows and the Effect precedes the causal marker.

ivvaLavu piraccinaikaL-ukk-um kaaraNam un aRiyaamai.
these-many problems-DAT-INC reason your ignorance.
"Your ignorance is the reason for all these problems."

5) toTarntu

This marker literally means "following which". So, this marker denotes consequence/cause.

uuraTaṅku uttarav-ai toTarntu terukkaL veRiccooTi irunt-ana.
Curfew order-ACC CAUSE streets empty be-PST-3PN.
"The streets were empty following the curfew order."

But this marker is polysemous. It can mean "regularly" or "continuously" or even "follow".

mantiravaati puuṅkuzaliyai toTarntu oot-in-aan.
Sorceror Poonkulali-ACC follow-VBP run-PST-3SM.
"The sorceror ran behind Poonkulali, following her"

6) atanaal/itanaal/aanapatiyaal/aakaiyaal/aatalaal

These markers are inter-clausal or inter-sentential markers meaning "so".They literally mean "because of that/this". Though they directly denote consequence, cause can be inferred.

ciRuvan tavaRu cey-t-aan. atanaal ammaa koopam-uR-R-aaL.
boy mistake do-PST-3SM. so mother anger-get-PST-3SF.
"The boy did a mistake. So, the mother got angry".

ciRuvan tavaRu cey-t-aan. itanaal ammaa avan-ai aTi-tt-aaL.
Boy mistake do-PST-3SM. because-of-this mother he-ACC beat-PST-3SF
"The boy did a mistake. So the mother beat him."

naańkaL pattirikkai vaańkuvat-illai. aanapatiyaal inta ceyti enakku teri-yaa-tu.
we newspaper buy-NEG. so this news I-DAT know-NEG-3SN.
"We don't buy newspapers. So, I don't know of this news."

en tantai-kku tamiz teri-yum. aakaiyaal avar-iTamiruntu tamiz kaR-kalaam.
my father-DAT Tamil know-3SN. so he-ABL Tamil learn-PERM
"My father knows Tamil. So, one can learn Tamil from him."

en aluvalaka neeram kaalai 11 maNi. aatalaal naan coompeeRi aaneen.
my office time morning 11o'clock. so I lazy become-PST-1S.
"My office time is at 11 a.m. So, I became lazy."

7) Verb in infinitive

This is a particular case of unmarked expressions of cause which is quite frequently found. The verb in the infinitive(morphologically) is used to chain a sequence of events, thus implicitly showing cause.

ciRuvan tavaRu cey-ya ammaa koopam-uR-R-aaL.
boy mistake do-INF mother anger-get-PAST-3SF.
"As the boy did a mistake, the mother got angry".

8) Verbs that denote cause

The following verbs may denote a causal relation in the sentence - *eeRpaTu, uNtaaku, viLai*.

cuuRaavaLi-y-aal peRum naacam viLai-nt-atu.
storm--CAUSE big damage lead-PST-3SN.
"The storm led to heavy damages."

9) Causative verbs

The causative verbs are a special class of verbs, where the additions of a marker (*-vi,-pi*) or an auxiliary verb (*vai, cey, aTi*) to the main verb produces another verb with the meaning "make to/cause to" added to the original meaning. The following examples show the use of auxiliary verbs to include causative meaning in the verb.

naan anta ceyti-y-ai aRi-nt-een.
I that news--ACC know-PST-1S.
"I knew that news."

naan anta ceyti-y-ai aRivi-tt-een.
I that news--ACC make know-PST-1S.
"I announced that news."

Here the causal interpretation is that "I am the cause for the news to be known."

naan anta paaTatt-ai avan-ukku puriya-vai-tt-een.
I that lesson-ACC he-DAT understand-make-PST-1S.
"I made him understand the lesson."

naan avan-ai caak-aTi-tt-een.
I he-ACC die-make-PST-1S.
"I caused him to die."

naan avaLai paaTa-c-cey-t-een.
I she-ACC sing--make-PST-1S.
"I made her sing."

IV. Computational representations

Computationally the above markers and examples can be expressed as Attribute Value Matrix (AVM) representations, which capture the arguments and features of the markers. The AVM representations for some of the examples are given in Figure 1.

1. `avar maaraTaippaal kaalamaanaar.`
   `He died of heart attack`

$$PP_{(CAUSE+aal)}\ \mathrm{Arg}(2)\quad \begin{bmatrix} \mathrm{Arg}_1 & [\mathrm{NP},\ -\mathrm{living},\ -\mathrm{concrete}] \\ \mathrm{Arg}_2 & \left[\mathrm{S}\ \begin{bmatrix} \mathrm{V} & \text{kaalamaaku arg}(1) \\ \mathrm{arg}_1 & [\mathrm{NP},\ +\mathrm{Sub},\ +\mathrm{Nom},\ +\mathrm{human},\ \pm\ \mathrm{male}] \end{bmatrix}\right] \end{bmatrix}$$

2. `kaaRRu aTittataal mazai peytatu.`
   `It rained because of the wind.`

$$PP_{(CAUSE+aal)}\ \mathrm{Arg}(2)\quad \begin{bmatrix} \mathrm{Arg}_1 & \left[\mathrm{S}\ \begin{bmatrix} \mathrm{V} & \text{aTi arg}(1) \\ \mathrm{arg}_1 & [\mathrm{NP},\ +\mathrm{Sub},\ +\mathrm{Nom},\ -\mathrm{living},\ -\mathrm{concrete}] \end{bmatrix}\right] \\ \mathrm{Arg}_2 & \left[\mathrm{S}\ \begin{bmatrix} \mathrm{V} & \text{pey arg}(1) \\ \mathrm{arg}_1 & [\mathrm{NP},\ +\mathrm{Sub},\ +\mathrm{Nom},\ -\mathrm{living},\ +\mathrm{concrete}] \end{bmatrix}\right] \end{bmatrix}$$

3. `takka neerattil maruttuvamanaikku cenRatanaal avar uyir pizaittaar.`
   `He was saved because he went to the hospital at the right time.`

$$PP_{(CAUSE+aal)}\ \mathrm{Arg}(2)\quad \begin{bmatrix} \mathrm{Arg}_1 & \left[\mathrm{S}\ \begin{bmatrix} \mathrm{V} & \text{cel arg}(2) \\ \mathrm{arg}_1 & [\mathrm{PRO},\ +\mathrm{Sub},\ +\mathrm{Nom},\ +\mathrm{living},\ \pm\ \mathrm{human}] \\ \mathrm{arg}_2 & [\mathrm{NP},\ +\mathrm{Obj},\ +\mathrm{Dat},\ -\mathrm{living},\ +\mathrm{concrete}] \end{bmatrix}\right] \\ \mathrm{Arg}_2 & \left[\mathrm{S}\ \begin{bmatrix} \mathrm{V} & \text{pizai arg}(2) \\ \mathrm{arg}_1 & [\mathrm{NP},\ +\mathrm{Sub},\ +\mathrm{Nom},\ +\mathrm{human},\ \pm\ \mathrm{male}] \\ \mathrm{arg}_2 & [\mathrm{NP},\ +\mathrm{Obj},\ +\mathrm{Nom},\ +\mathrm{part\_of\_body}] \end{bmatrix}\right] \end{bmatrix}$$

4. iRaiccalin kaaraNamaaka enakku onRumeee keetkavillai
   I cannot hear anything because of the noise.

$$ADV_{(CAUSE+kaaraNamaaka)}\ \mathrm{Arg}(2)$$
$$\begin{bmatrix} \mathrm{Arg}_1 & [\mathrm{NP},\ +\mathrm{Gen},\ \text{-concrete}] \\ \mathrm{Arg}_2 & \left[\ \mathrm{S}\ \begin{bmatrix} \mathrm{V} & \text{keeL arg(1)} \\ \mathrm{arg}_1 & [\mathrm{NP},\ +\mathrm{Sub},\ +\mathrm{Dat},\ +\mathrm{human},\ \pm\ \mathrm{male}] \end{bmatrix}\right] \end{bmatrix}$$

5. ivvaLavu piraccinaikaLukkum kaaraNam un aRiyaamai.
   The reason for all these problems is your ignorance.

$$NP_{(CAUSE+kaaraNam)}\ \mathrm{Arg}(2)$$
$$\begin{bmatrix} \mathrm{Arg}_1 & [\mathrm{NP},\ +\mathrm{Sub},\ \mathrm{Dat},\ \text{-concrete}] \\ \mathrm{Arg}_2 & [\mathrm{NP},\ +\mathrm{Obj},\ +\mathrm{Nom},\ \text{-concrete}] \end{bmatrix}$$

6. uuraTanku uttaravai toTarntu terukkaL veRiccooTi iruntana.
   The streets were empty following the curfew order.

$$PP_{(CAUSE+toTarntu)}\ \mathrm{Arg}(2)$$
$$\begin{bmatrix} \mathrm{Arg}_1 & [\mathrm{NP},\ +\mathrm{Acc},\ \text{-concrete}] \\ \mathrm{Arg}_2 & \left[\ \mathrm{S}\ \begin{bmatrix} \mathrm{V} & \text{iru arg(1)} \\ \mathrm{arg}_1 & [\mathrm{NP},\ +\mathrm{Sub},\ +\mathrm{Nom},\ \text{-living},\ +\mathrm{concrete}] \end{bmatrix}\right] \end{bmatrix}$$

Φιγυρε 1. Some example AVMs

V. Conclusion

This attempt at the analysis of the cause-effect semantic relation in Tamil and the AVMs can be used in automatic identification of causal relations in text. This, in turn, would be useful in information retrieval systems and reasoning or question-answering systems.

References

[1] D. Ehring, Causation and persistence: A theory of causation. New York: Oxford University Press, 1997.

[2] R. Girju, "Automatic Detection of Causal Relations for Question Answering." In the proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003), Workshop on "Multilingual Summarization and Question Answering - Machine Learning and Beyond", 2003.

[3] C. S. G. Khoo, "Automatic identification of causal relations in text and their use for improving precision in information retrieval." Ph.D. dissertation, School of Information Studies, Syracuse University, 1995.

[4] C. Khoo, S. H. Myaeng and R. Oddy, "Using cause-effect relations in text to improve information retrieval precision". Information Processing and Management, 37(1), pp. 119-145, 2001.

[5] C. Khoo, and S. H. Myaeng, Identifying semantic relations in text for information retrieval and information extraction. In R.Green, C.A. Bean & S.H. Myaeng (Eds.), The semantics of relationships: An interdisciplinary perspective (pp. 161-180). Dordrecht: Kluwer, 2002.

[6] C. Khoo, S. Chan, and Y. Niu, "The many facets of the cause-effect relation." In R. Green, C. Bean and S. H. Myaeng, The semantics of relationships: An interdisplinary perspective (pp. 51-70). Dordrecht: Kluwer, 2002.

[7] D. H. Mellor, The facts of causation. London: Routledge, 1995.

[8] A. Nazarenko, "Representing Natural Language Causality in Conceptual Grpahs: the Higher Order Conceptual Relation Problem." In Proceedings on Conceptual Graphs For Knowledge Representation (August 04 - 07, 1993). G. W. Mineau, B. Moulin and J. F. Sowa, Eds. Lecture Notes In Computer Science, vol. 699. Springer-Verlag, London, pp. 205-222, 1993.

[9] D. Owens, Causes and coincidences. Cambridge: Cambridge University Press, 1992.

[10] E. Sosa and M. Tooley, Eds.), Causation. Oxford: Oxford University Press, 1993.

Named Entity Recognition and Transliteration for Telugu Language

Kommaluri VIJAYANAND and R. P. Seenivasan
Department of Computer Science
School of Engineering and Technology
Pondicherry University
Puducherry – 605 014, India.
Email: kvixs@yahoo.co.in, rpsv@yahoo.com

1. Introduction

The concept of transliteration is a wonderful art in Machine Translation. The translation of named entities is said to be transliteration. Transliteration should not be confused with translation, which involves a change in language while preserving meaning. Transliteration performs a mapping from one alphabet into another. In a broader sense, the word transliteration is used to include both transliteration at the micro level and transcription.

Transliteration is a process in which words in one alphabet are represented in another alphabet. There are a number of rules which govern transliteration between different alphabets, designed to ensure that it is uniform, allowing readers to clearly understand transliterations. Transliteration is not quite the same thing as transcription, although the two are very similar; in transcription, people represent sounds with letters from another alphabet, while in transliteration, people attempt to map letters over each other, sometimes with accent marks or other clues to suggest particular sounds.

As we say technically the transliteration is the process of transforming the text in one writing system (Source language) to another writing system (Target Language) without changing its pronunciation. Transliteration is a very good asset for machine translation. Machine translation cannot translate some of the text. Because, there could not be correspond translation word in the bilingual dictionary. Those words are called out of vocabulary words (OOV). To overcome this OOV problem transliteration came into being. The transliteration involves the process of converting the character sequence in the source language to target language on the basis of how the characters are pronounced in source language.

Transliteration needs knowledge of characters in source and target language. Since the pronunciation is the aim goal of transliteration it is difficult to give exact transliteration. Because, the pronunciation of single character of the source language can have multiple character in the target language as the transliteration is done by character wise. In transliteration so far we can give possible transliterations and yet it is the great challenge to the researchers to give exact transliteration in target language.

People try to use standardized trends when they transliterate so that transliterations are uniform, but this does not always happen. Muhammad, for example, is spelled in a variety of ways, including Mohammad and Mahomet. This can be confusing, as changes in transliteration change the way that a word sounds when it is spoken out loud. A good transliteration also employs accent markings to guide people, and it may have unusual combinations of letters in an attempt to convey unique sounds. Transliteration is not the same thing as translation, a process in which

words are made meaningful to speakers of other languages. Translation requires knowledge of languages, where transliteration is more about alphabets.

2. The Origin of the System

Advances information technology leads to the discovery of transliteration. Today transliteration plays a major role in all aspects of the society. There are a number of reasons to use transliteration, but most of them involve conveying information across cultures. Transliteration is needed in our day – to- day life. Even translation cannot be fulfilled without this translation. The translation of named entities cannot be possible in machine translation. In every writings named entities play a major role. So without named entities a text cannot be fulfilled. The named entities can be transliterated and cannot be translated. So the translation system also needs transliteration.

We can explain the use of transliteration using an example. For example when a Telugu man who don't know to read English going to restaurant, if he see menu card which is in English he can't order anything because of his lack of English reading knowledge. Suppose the menu card consists of Telugu transliteration of those menus he can order the food items without knowing what it is.

In literature also transliteration plays a role. When the translator translates the novels or stories they need transliteration in case names of persons and places. Transliteration is also used in language education, so that people can understand how words are pronounced without needing to learn the alphabet as well. Academic papers may also use transliteration to discuss words in various languages without forcing their readers to learn an assortment of alphabets.

In the Internet also the transliteration is applied. Usually the web news is all in English. When we need it in any other language the websites has the facility to display it in that particular language. In that translated web page out of vocabulary words are transliterated.

In the natural language processing applications such as machine translation, cross language information retrieval, question answering system etc., the transliteration is used.

Initially there is a technical motivation of building intelligent computer system such as Machine Translation (MT) systems, natural language (NL) interfaces to database, man-machine interfaces to computers in general, speech understanding system, text analysis and understanding systems, computer aided instruction systems, system that read and understand printed or hand written text. Second, there is a cognitive and linguistic motivation to gain a better insight into how humans communicate using natural language.
For development of any natural language processing system, there are several sources of knowledge that are used in decoding the information from an input. These can be classified as follows:-

- Language knowledge
    (a) Grammar
    (b) Lexicon
    (c) Pragmatic and discourse. Etc.
- Background Knowledge

(a) General world knowledge (including common sense knowledge)
(b) Domain specific knowledge (includes specialized knowledge of the area about which communication is taking place)
(c) Context (Verbal or non-verbal situation in which communication is to take place)
(d) Cultural knowledge

From the various sources of knowledge mentioned above, a hearer (or a reader) can extract information conveyed form a given source (a speaker or writer).

3. The Methodology

In Grapheme-Based method, source words are transcribed to the target words based on grapheme units directly without making use of their phonetic representations. The grapheme based method is called direct method. The grapheme based technique is direct orthographical mapping from source graphemes to target graphemes.

The methods based on the source-channel model deal with English-Telugu transliteration. They use a chunk of graphemes that can correspond to a source phoneme. First, English words are segmented into a chunk of English graphemes. Next, all possible chunks of Telugu graphemes corresponding to the chunk of English graphemes are produced. Finally, the most relevant sequence of Telugu graphemes is identified by using the source-channel model. The advantage of this approach is that it considers a chunk of graphemes representing a phonetic property of the source language word. However, errors in the first step (segmenting the English words) propagate to the subsequent steps, making it difficult to produce correct transliterations in those steps. Moreover, there is high time complexity because all possible chunks of graphemes are generated in both languages. In the method based on a decision tree, decision trees that transform each source grapheme into target graphemes are learned and then directly applied to machine transliteration. The advantage of this approach is that it considers a wide range of contextual information, say, the left three and right three contexts.

Furthermore, they segment a chunk of graphemes and identify the most relevant sequence of target graphemes in one step. This means that errors are not propagated from one step to the next, as in the methods based on the source-channel model. The method based on the joint source-channel model simultaneously considers the source language and target language contexts (bigram and trigram) for machine transliteration. Its main advantage is the use of bilingual contexts.

3.1. The Algorithm

The present transliteration system is implemented using the algorithm narrated step wise as follows:

1. The input for this system is an xml file.
2. This xml file consists of only names in source language.
3. The xml file is read and the source names are extracted and stored in the array list.
4. Source names are retrieved from the array list one by one for the further process.
5. Then the source name is rewritten using rewriting Techniques.

6. The next step is segmentation
7. After segmentation the chunks retrieved from the array list where they are stored one by one for target grapheme retrieval
8. In the target grapheme collection process the source grapheme is compared with the database and all the relevant graphemes are collected and stored it in the array list
9. The target graphemes of first grapheme is stored in one array list and that target graphemes of other source graphemes are stored in one array list.
10. After generation of target names for the source names and it is stored in the xml file.

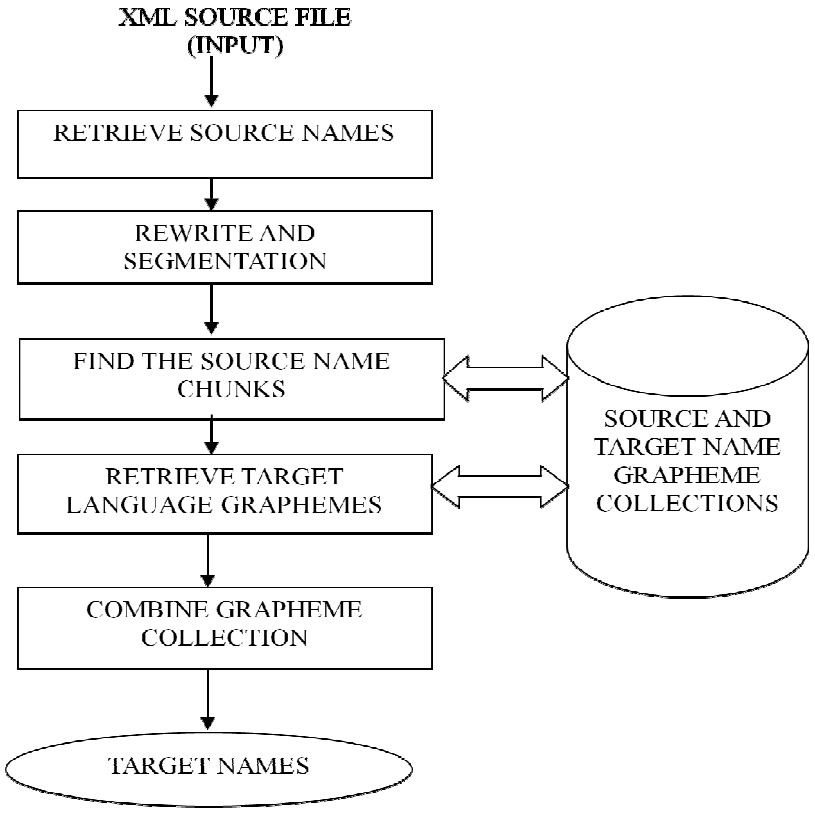


Figure 1: Block Diagram of the Machine Transliteration System

4. Implementation details

The System has been designed and implemented in java using swings for interface that takes various input queries from user and outputs the translated query in Telugu. The internal interaction and working of the system has been implemented using Java. The coding phase aims at translating the design of the system into code. The code has then been executed and tested. The goal is to implement the design in the best possible manner.

Rule-Based method

It requires analysis and representation of the meaning of source language texts and the generation of equivalent target language texts. Representation should be unambiguous lexically and structurally. There are two major approaches:

- The transfer approach in which translation process operates in three stage-analysis into abstract source language representations, transfer into abstract target language representations and generation or synthesis into target language text.
- The two stage 'interlingua' model where analysis into some language-neutral representation starts from this Interlingua representation.

Source Name Retrieval

The input for this system is an xml file. This xml file consists of only names in source language. The xml file is read and the source names are extracted and stored in the array list. Source names are retrieved from the array list one by one for the further process.

Rewrite and Segmentation

There are several rules and methods for rewriting and segmentation. Some of such rules are listed as follows:

- If the second index to the current index of the word is 'a' or 'e' or 'I' or 'o' or 'u' then it is considered as one segment.
- If the second index to the current index of the word is 'h' and the third index to the current index of the word is 'a' or 'e' or 'I' or 'o' or 'u' then it is considered as one segment.
- If the second and third index to the current index of the word is 'a' or 'e' or 'I' or 'o' or 'u' and it is same character i.e. 'aa', 'ee' ,'oo' then is considered as one segment.

- If the second index to the current index of the word the word 'a' , 'o' and the third index to the current index of word is 'e', 'u' then it is considered as one segment.
- If the second and third index to the current index of the word does not satisfy the above four conditions then the current index of the word is considered as one segment.
- After segmentation, the graphemes of source name (English) are compared with the database and target graphemes are collected.

After collecting target graphemes those graphemes merged to generate transliterations in target language (Telugu).

Source Name Chunks

This method was applied with the rule based algorithm. This algorithm is based on translating the linguistic rules into machine readable form. These rules are hand-crafted.

- The first step in the implementation is to rewrite the name.
- This step is used to reduce the unnecessary occurrence of 'h', repeated characters, and replace the characters having the same sound.

Retrieval of Target Language Graphemes

There are several handcrafted rules for rewriting process of named entities. They are:

- The next step in the algorithm is Segmentation.
- The segmentation is also done on the basis of handcrafted rules.

Segmentation is done with the rules as said before. In the segmentation process the names are segmented in to chunks using those rules and are stored in an array list. After segmentation the chunks retrieved from the array list where they are stored one by one for target grapheme retrieval. In the target grapheme collection process the source grapheme is compared with the database and all the relevant graphemes are collected and stored it in the array list. The target graphemes of first grapheme is stored in one array list and that target graphemes of other source graphemes are stored in one array list. The value of second array list is merged with first array list. The value of second array list changed dynamically. After generation of target names from the source names it will is stored in the xml file.

Conclusion

Based on the techniques and methods used to transliterate the named entities from English to Telugu language, we had found that for writing system comprises of the graphemes and phonemes that play major role in transliteration. The writing system for both the Tamil and Telugu languages is same and share common properties during transliteration system development. Thus application of Machine Learning would help in developing a common generator with different production algorithms based on the South Indian Languages like Kannada, Malayalam, Telugu and Tamil.

References:

[1]. Eduard Hovy and Chin-Yew Lin, Automated Text Summarization in SUMMARIST, In Advances in Automatic Text Summarization, 1999.

[2].George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller**,** Introduction to WordNet**,** 1993.

[3]. Harshit Surana and Anil Kumar Singh, A More Discerning and Adaptable Multilingual Transliteration Mechanism for Indian Languages, Proceedings of International Joint Conference on Natural Language Processing, Hyderabad, India, 2008.

[4]. Surya Ganesh, Sree Harsha, Prasad Pingali, and Vasudeva Varma., Statistical Transliteration for Cross Language Information Retrieval using HMM alignment and CRF, Proceedings of International Joint Conference on Natural Language Processing(_CNLP)-2008, NERSSEAL Workshop, Hyderabad, India, 2008.

[5]. The Unicode Standard version 3.0 (http://www.unicode.org)

[6]. An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Second edition by Daniel Jurafsky, James H.Martin

[7]. T. Rama and K. Gali, Modeling machine transliteration as a phrase based Statistical Machine Translation Problem, In proceedings of the Named Entities Workshop, ACL–IJCNLP 2009, pp. 124-127, August 2009.

[8]. Nayan , B. R. K. Rao, P. Singh, S. Sanyal, and R. Sanyal, "Named entity recognition for Indian languages," In Proceedings of International Joint Conference on Natural Language Processing (IJCNLP), pp. 97-104, 2008.

[9]. N. A. Jaleel and L. S. Larkey, "Statistical transliteration for english-arabic cross language information retrieval," In Proceedings of the twelfth international conference on Information and knowledge management, November 03-08, 2003, New Orleans, LA, USA.

Identification of Different Feature Sets for NER tagging using CRFs and its impact

Vijay Sundar Ram R and Pattabhi R.K. Rao and Sobha Lalitha Devi
AU-KBC Research Centre
MIT Campus of Anna University
Chennai, India

***Abstract*- This paper presents a study of the impact of different types of language modeling by selecting different feature matrices in the Conditional Random Fields (CRFs) learning algorithm for Named Entity tagging. We have come up with four different feature matrices and identified features at word, phrase and sentence level. It is identified that the language model which has the structural feature is better than the models with other features.**

I. INTRODUCTION

In this paper, we present a study on how the performance of the Named Entity Recognition (NER) using Conditional Random Fields (CRFs) varies according to different features and feature matrices. Named Entity tagging is a labeling task. Given a text document, named entities such as Person names, Organization names, Location names, Product names are identified and tagged. Identification of named entities is important in several higher language technology systems such as information extraction, machine translation systems.

Named Entity Recognition was one of the tasks defined in MUC 6. Several techniques have been used for Named Entity tagging. A survey on Named Entity Recognition was done by David Nadaeu[6]. The techniques used include rule based technique by Krupka [9], using maximum entropy by Borthwick [4], using Hidden Markov Model by Bikel [3] and hybrid approaches such as rule based tagging for certain entities such as date, time, percentage and maximum entropy based approach for entities like location and organization [16]. There was also a bootstrapping approach using concept based seeds [14] and using maximum entropy markov model [7]. Alegria et al, [1], have developed NER for Basque, where NER was handled as classification task. In their study, they have used several classification techniques based on linguistic information and machine learning algorithms. They observe that different feature sets having linguistic information give better performance.

Lafferty [11] came up with Conditional Random Fields (CRFs), a probabilistic model for segmenting and labeling sequence data and showed it to be successful with POS tagging experiment. Sha and Pereira [17] used CRFs for shallow parsing tasks such as noun phrase chunking. McCallum and Li [12] did named entity tagging using CRFs, feature induction and web enhanced lexicons. CRFs based Named Entity tagging was done for Chinese by Wenliang Chen [21]. CRFs are widely used in biological and medical domain named entity tagging such as work by Settles [18] in biomedical named entity recognition task and Klinger's [8] named entity tagging using a combination of CRFs. The Stanford NER software [10], uses linear chain CRFs in their NER engine. Here they identify three classes of NERs viz., Person, Organization and Location. Here they have used distributional similarity features in their engine, but this utilizes large amount of system memory. This paper discusses different feature sets used and their impacts in CRFs for NER.

The paper is further organized as follows. In Section 2 we have described our approach for identifying the suitable feature matrix. Section 3 presents the different experiments, results obtained and discussion on the performance of each experiment. Section 4 concludes the paper.

II. OUR APPROACH

In this work we have used a machine learning technique for identification of named entities. Here we did four different experiments by varying the feature matrix given to the training algorithm of the machine learning approach to study the performance and to choose the best feature set for identifying the named entities.

We have used Conditional Random Fields (CRFs) for the task of identifying the named entities. CRFs is undirected graphical model, where the conditional probabilities of the output are maximized for a given input sequence. CRFs is one of the techniques best suited for sequence labeling task. Hidden Markov Model (HMM), Maximum Entropy Markov Model (MEMM) and CRFs are well suited for sequence labeling task. MEMM and CRFs allows linguistic rules or conditions to be incorporated into machine learning algorithm. HMM [15] does not allow the words in the input sentence to show dependency among each other. MEMM [2] shows a label bias problem because of its stochastic state transition nature. CRFs, overcomes these problems and performs better than the other two.

A. Conditional Random Fields

CRFs make a first order Markov independence assumption and can be viewed as conditionally trained probabilistic finite state automata.

Now let $o=(o_1,\ldots,o_T)$ be some observed input data sequence, such as a sequence of words in a text document, (the values on T input nodes of the graphical model). Let S be a set of FSM states, each of which is associated with a

label, $l \in L$, (such as PERSON). Let $s=(s_1,\ldots,s_T)$ be some sequence of states, (the values on T output nodes).

Linear-chain CRFs thus define the conditional probability of a state sequence given as follows

$$P_\Lambda(\mathrm{s}|\mathrm{o}) = \frac{1}{Z_\mathrm{o}} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k f_k(s_{t-1}, s_t, \mathrm{o}, t)\right),$$

where Z_0 a normalization factor over all state sequences, $f_k(s_{t-1},s_t,o,t)$ is an arbitrary feature function over its arguments, and λ_k (ranging from $-\infty$ to ∞) is a learned weight for each feature function. A feature function may, for example, be defined to have value 0 in most cases, and have value 1 if and only if s_{t-1} is state #1 (which may have label OTHER), and s_t is state #2 (which may have PERSON or PRODUCT or TITLE label), and the observation at position t in o is a proper noun or an adjective. Higher λ weights make their corresponding FSM transitions more likely, so the weight λ_k in the above example should be positive since the word appearing is any NE category (such as LOCATION or PRODUCT-COMPONENT) and it is likely to be the starting of a named entity.

We have used an open source toolkit for linear chain CRFs called as CRF++ [19].

B. Feature Matrix

The training of the CRFs requires iterative scaling techniques, where a quasi-Newton method such as L-BFGs is used. The input data for the task is processed with part-of-speech tagging (POS) and chunking. Part-of-Speech tagging is done using the Brill's Tagger [5] and text chunking is done using fn-TBL [13]. In the shallow processed text, named entities are manually annotated using a NE tagset, containing Person, Organization, Location, Facility, Product, Product Component, Output, and Output Quality as tags. This processed data is used as the training data.

The choice of features is as important as the choice of technique for obtaining a good Named Entity Recognizer [20]. The selection of the set of feature can improve the results. Here we have presented the NE annotated input in four different forms of feature matrix.

Type1

The complete sentence is represented in the feature matrix. Consider the sentence in the example 1.

(1) "I love point-and-shoots and have no desire at this point to get DSLR".

The feature matrix for this type 1 would be as shown below.

Feature Matrix of Type 1, for example 1

I	PRP	B-NP	B-PERSON
love	VBP	B-VP_act	o
point-and-shoots	NNS	B-NP	o
and	CC	o	o
.	.	.	.
.	.	.	.
get	VB	I-VP	o
DSLR	NN	B-NP	B-PRODUCT
.	.	o	o

Type 2

The feature matrix for the second type is built by taking only the Noun Phrases (NPs) from the training data. From the example 1, we obtain six sequences, because there are six noun phrases in this sentence. A sample of the feature matrix of type 2 is shown below.

Feature Matrix of Type 2, for example 1

I	PRP	B-NP	B-PERSON
point-and-shoots	NNS	B-NP	o
no	DT	B-NP	o
desire	NN	I-NP	o
this	DT	B-NP	o
desire	NN	I-NP	o
this	DT	B-NP	o
point	NN	I-NP	o
DSLR	NN	B-NP	B-PRODUCT

Type 3

Named Entities (NEs) with one preceding word and one following word, is considered from the processed input text to build the feature matrix for the third type. A window of size three is taken. Considering the example 1, we have two named entities, 'I', which has PERSON, NE tag and 'DSLR', having PRODUCT, NE tag. Here for this example we obtain two sequences in the feature matrix as shown below.

Feature Matrix of Type 3, for example 1

:	:	o	o
I	PRP	B-NP	B-PERSON
love	VBP	B-VP_act	o
get	VB	I-VP_act	o
DSLR	NN	B-NP	B-PRODUCT
,	,	o	o

Type 4

In this type, a window of size five is considered. NEs is taken along with two preceding and two following words. Here we provide more contextual information by increasing the size of the window to five. Considering the sentence in example 1, the feature matrix consists of two sequences, where each sequence has one more word added to the left and right of the NE, comparing to the feature matrix of type 3. The sample of feature matrix of type 4 is shown below.

Feature Matrix of Type 4, for example 1

camera NN B-NP I-TITLE
: : o o
I PRP B-NP B-PERSON
love VBP B-VP_act o
point-and-shoots NNS B-NP o
to TO B-VP_act o
get VB I-VP_act o
DSLR NN B-NP B-PRODUCT
, , o o
and CC o o

C. Features of CRFs

The set of features used are word, phrase and structure level. The word level features are words or tokens that occur in the first column of the feature matrix. The word level features are current word, previous to previous word, previous word, next word, next to next word.

Phrase level features include words, POS tags and chunk or phrase information. Phrase level or chunk level features are

(a) current word's POS and chunk information,
(b) current word's POS,
(c) previous word's POS and next word's POS.

The following are sample rules learnt by CRF from phrase level features

Rule P1:

-1 w1 DT
0 w2 NNP PRODUCT

This rule describes if the previous word's POS is determiner (DT) and current word has POS tag as 'NNP' then the current word is tagged as PRODUCT

Rule P2:

-1 w1 JJ NP OUTPUT-QUALITY
0 w2 NN NP OUTPUT-QUALITY

The above rule describes if the previous word's POS is adjective (JJ) and current word's POS is 'NN' then both current word and the previous word will be tagged as 'OUTPUT-QUALITY'.

Structure level features includes features such as

i) Current word given the Current word's POS tag
ii) and Previous word,
iii) Current word given the previous word and its chunk information
iv) Current word given the next word and its POS tag,
v) Current word's chunk information given previous word and its chunk information.
vi) Current word's POS tag given the previous word's NE tag and the next word's NE tag.

Here we consider these to be dependent on each other and find the conditional probabilities. The sample rules learned by CRF engine from the structural features are described below.

Rule S1:

-1 consumes VP
0 w1 NP OUTPUT-QUALITY

The above rule describes if the previous word is 'consumes' which is a verb phrase (VP) then the current word which is a noun phrase (NP) will be tagged as OUTPUT-QUALITY.

Rule S2:

-2 rate/rates/rated VP
-1 w1 NP PROD-COMP
0 w2 NP OUTPUT-QUALITY

The above rule describes if the previous to previous word is 'rate' or 'rates' or 'rated', which is a VP and if the previous word is a NP having the NE tag as Product Component (PROD-COMP) then the current word which is a NP will be tagged as OUTPUT-QUALITY

Rule S3:

-2 w1 NP PERSON
-1 purchased VP
0 w2 NP PRODUCT

If the previous to previous word is a NP with NE tag as PERSON and the previous word is 'purchased' which is a VP then the current word which is a NP will be tagged as PRODUCT.

Using the feature matrices built from input training sentences. The different language models are built by CRFs training algorithm.

Thus the language model LM1 is built from feature matrix of Type1. Here the model learns the structure of the complete sentence, both the structures, where NEs and non- NEs occur. The occurrence of non NEs is more. The language model LM2 is built by training the feature matrix formed by type 2. Here the NEs occurring inside the NPs are learned and rest are not seen by the CRF engine. Using the feature matrix built by type 3, which contains a sequence of window of size three, language model LM3 is formed. This has contextual information of the NEs. The language model LM4 is built using the feature matrix of type 4, which is formed using NE, with a window of size five.

We have performed four different experiments, to study how the performance of the named entity recognizer varies when different language models and different features are used. The experiments and their results are described in the following section.

III. Experiments, Results and Discussion

The training data consists of 1000 documents describing user reviews on different electronic goods such as mobiles, camcorders, notebooks. These documents were obtained from

online trading websites such as Amazon, eBay. The training data consisted of 3107 unique NEs and the number of occurrence of NEs is 24345. The test data consists of similar type of documents. This consists of 456 non-unique NEs. This test data consisted of 94 NEs which were not in the training data. This constitutes 20.6% out-of vocabulary words (OOV words). The test data consisted of 300 unique NEs. The 94 not seen NEs had no repetition, they were unique. Here we have performed CRFs training using the four different feature matrices to build different language models. In the first experiment, language model LM1, is built using CRFs by taking the full sentences in the training data as sequences. The LM1 is taken as the baseline language model. The second experiment, language model LM2 is built by taking the Noun Phrases (NPs) as sequences. In the third experiment, language model LM3 is built by taking NE, with a window of three. In the fourth experiment, language model LM4 is built by NE, with a window of five. The table 1 below, show the results obtained, by doing NE identification on test data using the four different language models.

As we observe the results, in the LM1 model, the learning algorithm learns many rules, from the training data, this makes an overfit, due to which false positives is more and hence the precision is less. In the LM2 model, even though the number of false positives is reduced, and the precision increases slightly, the recall does not increase significantly. In this model, the disambiguation of the NE tags is poor, the learning algorithm does not get any context information, since only NPs are presented to the learning algorithm during training. This does not handle the OOV words. In the LM3 model, we observe that the precision and recall increase significantly. Since in this model the NEs and the preceding and following words are presented to the learning algorithm during training, this gives contextual information, and this learns only the structure of the sentence, where NE can occur. This reduces the number of irrelevant rules, which confuses the learning algorithm. So we obtain better results compared to the first two models. In LM4, we introduce more contextual information to the feature matrix, by considering a window of size five. This helps in learning the structure of the sentence, where the NEs occur more precisely, which increases the recall. Since the feature matrix has the NE with a window of five, more relevant rules are learnt by the system. This reduces the false positives and increases the true positives. The precision increases. As in this model, the structure of sentence, where NE occurs, OOV words also identified. The recall in this model also increases. Hence we obtain a Precision of 96.4% and Recall of 90.1%. The F-measure for this LM4 model is 93.14%.

A. Role of Different Features in Learning

In the table 2 below, the results obtained on using different set of features while learning for the LM4 model are shown.

The word level features and chunk level features help in obtaining rules based on the syntactic information in sentence. The structural features help in learning the sentence structures, where the NEs can occur in a sentence. As this task of NE identification does not require learning of the complete structural information of the sentence.

As we observe in the table 2, when word level features alone are used, the precision is high, but not the recall, because, here the algorithm does not learn sentence structures, and is completely dependent on the words. Hence does not handle out-of-vocabulary (OOV) words. In practice, the real time data consists of OOV words. When we use chunk level features along with the word level features, it is observed that the precision decrease, but the recall increases significantly. This can be explained by the fact that, using chunk level feature, makes the learning algorithm to infer from the POS tags and chunk information, and not just the words alone.

When the structural features are used along with the word and chunk level features, we find that the recall increase significantly, without deteriorating the precision. When the structural features are used, the conditional probabilities calculated are considering the context of the NE, hence this creates a context based model, and makes the CRFs learn the sentence structure well. This in turn helps in handling the 60% of the OOV words.

In the table 3, we find that two NE classes Output and OutputQuality have less recall compared to other NE classes. The occurrence of the NE tags 'Product components', 'Output' and 'OutputQuality' are more ambiguous. For example Nokia N73 is a product NE and its feature such as wifi, 4 mega pixel. mp3 player are tagged as the 'Output' and in the case of a camera, 4 mega pixel is tagged as a Product component. This creates ambiguity, while building the training model. Also these NEs, does not occur enough number of times for the CRFs to learn well. This affects the recall. It was also observed that for the tags Product-Component and Output, the inter annotator agreement is low. This resulted in recall and precision to be lower for both these NE classes. This shows how the inter annotator agreement affects the performance of named entity recognizer.

TABLE I
RESULTS ON USING DIFFERENT LANGUAGE MODELS

Models	Total NEs	NEs Identified	NEs Identified correct	Precision (%)	Recall (%)	F-Measure (%)
LM1	456	388	348	89.7	76.3	82.46
LM2	456	397	362	91.2	79.3	84.83
LM3	456	402	375	93.3	82.2	87.39
LM4	456	426	411	96.4	90.1	93.14

TABLE II
ROLE OF DIFFERENT SET OF FEATURES

Features taken	Total NEs	NEs identified	Correct NEs	Precision (%)	Recall (%)	F-Measure (%)
Word Level	456	320	315	98.4	69.1	81.19
+Chunk level	456	380	352	92.6	77.2	84.20
+Structural Features	456	426	411	96.4	90.1	93.14

TABLE III
NE TAG WISE RESULT BY USING LM4 MODEL

NE Tag	Total NEs	NEs Identified	Correct NEs	Precision (%)	Recall (%)
Person	77	75	73	97.3	94.8
Product	122	115	111	96.5	90.9
Product Component	143	137	133	97.1	93.0
Output	54	48	45	93.7	83.3
Output Quality	60	51	49	96.1	81.7

IV. CONCLUSION

In this work we study the performance of the NE identification task using CRFs by building four different language models by varying the feature matrix constructed from the NE annotated and preprocessed input sentences. The language model, LM4, NEs with a window of five, performs the best of all four. We obtain an F-measure of 93.14%.

We have performed experiments to study the impact of various features on the performance of the NER. Here we have selected three different types of features, word level, chunk or phrase level and structural level. We identify that the best performance is obtained when all the three types of features are used together in learning. If only word level features are used, NER does not handle OOV words, when both chunk level and word level features are used, the learning algorithm does not learn the sentence structures effectively.

We also observe the how the inter annotator agreement plays a vital role in the performance of the NERs using CRFs. It is observed that when the inter annotator agreement is low, the training data consists of ambiguous tagging and this creates ambiguity for the learning algorithm. Hence the performance gets negatively affected.

REFERENCES

[1] Alegria I, Arregi O, Ezeiza N, Fernandez I. Lessons from the Development of Named Entity Recognizer for Basque, *Natural Language Processing*, 36,2006. pp. 25 – 37.

[2] Berger A, Della Pietra S and Della Pietra V. A Maximum Entropy Approach to Natural Language Processing, *Computational Linguistics*, 22(1), 1996

[3] Bikel D M. Nymble: a high-performance learning name-finder, *In Proceedings of the Fifth Conference on Applied Natural Language Processing*, 1997.pp.194-201.

[4] Borthwick A, Sterling J, Agichtein E and Grishman R. Description of the MENE named Entity System, *In Proceedings of the Seventh Machine Understanding Conference (MUC-7)*, 1998.

[5] Brill, Eric. Some Advances in transformation Based Part of Speech Tagging, *In the Proceedings of the Twelfth International Conference on Artificial Intelligence (AAAI- 94)*, Seattle, WA, 1994.

[6] Nadeau, David and Satoshi Sekine (2007) A survey of named entity recognition and classification. *Linguisticae Investigationes* 30(1):3–26.

[7] Jenny Finkel, Shipra Dingare, Huy Nguyen, Malvina Nissim, Gail Sinclair and Christopher Manning. Exploiting Context for Biomedical Entity Recognition: from Syntax to the Web. *In the Proceedings of Joint Workshop on Natural Language Processing in Biomedicine and its Applications, (NLPBA)*, Geneva, Switzerland, 2004.

[8] Roman Klinger, Christoph M. Friedrich, Juliane Fluck, Martin Hofmann-Apitius. Named Entity Recognition with Combinations of Conditional Random Fields. *In Proceedings of 2nd Biocreative Challenge Evaluation Workshop*, CNIO, Madrid,Spain, 2007.pp. 89-92

[9] Krupka G R and Hausman K. Iso Quest Inc: Description of the NetOwl Text Extraction System as used for MUC-7. *In Proceedings of Seventh Machine Understanding Conference (MUC 7)*, 1998.

[10] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *In the proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, 2005 pp. 363-370.

[11] John Lafferty, Andrew McCallum, Fernando Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *In Proceedings of the Eighteenth International Conference on Machine Learning (ICML-2001)*, 2001.pp.282-289.

[12] Andrew McCallum and Wei Li. Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons, *In Proceedings of Seventh Conference on Natural Language Learning (CoNLL)*, 2003.

[13] G. Ngai and R. Florian. Transformation-Based Learning in the Fast Lane, *In the Proceedings of the NAACL'2001*, Pittsburgh, PA, 2001.pp.40-47

[14] C. Niu, W. Li, Rohini K. Srihari. Bootstrapping for Named Entity Tagging using Concept-based Seeds. *In Proceedings of HLT-NAACL'03, Companion Volume*, Edmonton, 2003.pp.73-75.

[15] Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *In Proceedings of the IEEE*, 77(2), 1989.pp.257– 286.

[16] Rohini K Srihari, C.Niu and W.Li. Hybrid Approach for Named Entity and Sub-type Tagging. *In Proceedings of Applied Natural Language Processing Conference*, Seattle, 2000.pp.247-254.

[17] Fei Sha and Fernando Pereira. Shallow Parsing with Conditional Random Fields, *In the Proceedings of HLT-NAACL 03*, 2003. pp.213-220

[18] Settles B. Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets. *In Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA)*, Geneva, Switzerland, 2004.pp.104-107.

[19] Taku Kudo. CRF++, an open source toolkit for CRFs, http://crfpp.sourceforge.net, 2005.

[20] Tjong Kim Sang, Erik. F.; De Meulder, F. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. I*n Proceedings of Conference on Natural Language Learning*. 2003

[21] Wenliang Chen, Yujie Zhang and Hitoshi Isahara. Chinese Named Entity Recognition with Conditional Random Fields. *In Proceedings of Fifth SIGHAN Workshop on Chinese Language Processing*, Sydney, 2006.pp.118-121.