A Study of Mathematical Morphology of Gujarati Script using Wavelet Optimization and Soft Computing Techniques

A Ph.D. Synopsis
Submitted to
Gujarat Technological University, Chandkheda

In Partial fulfilment
For the Award of
Ph. D. in Maths-Science

by

Manisha Prajapati
Enrolment No: 149997673007

Under supervision of

Dr. Archit Yajnik,
Professor and Head,
Sikkim Manipal University

Co-Supervisor:
1. Dr. Trupti Shah, M.S. University, Vadodara

DPC Members:
1. Dr. S. Rammohan, IITRAM, Ahmedabad
2. Dr. Jayesh Dhodiya, SVNIT, Surat.

Index

1. Abstract

Part of Speech (POS) Tagging and Parsing play a vital role in the applications like Machine translation of Natural Language Processing. Significant research articles [35] are available in the literature for Indian languages however Gujarati is least [CRF model] explored as far as POS tagging and parsing is concern. The present work is mainly focusing on these two aspects of NLP viz.

POS tagger using Viterbi algorithm where bigram model of Hidden Markov Model is implemented in matlab using the corpus of 1700 annotated Gujarati words. The database is generated manually with the concertation of linguistic. Tag set of BIS (Bureau of Indian standards) is used during the experiment. The tagger has achieved the accuracy of 92.87% [4]. In order to gain better result Artificial Neural Network based Support Vector Machine is implemented. The feature vector for each word is generated using transition and emission probability which is supplied as an input neuron to the network and the vector of 28 neurons, for each tag, is taken as output layer. In order to minimize the training time the dimension of input vector is reduced by supplying Daubechies D4 wavelet based compressed features to the SVM network. The network took significantly lesser training time as compared to earlier SVM tagger but only 58% accuracy is reported.

The constraint formation based on Paninian framework to develop shallow parser is introduced by Akshar Bharati and Rajeev Sangal in [8]. Two techniques are demonstrated in the article viz Linear Programming approach and Maximum Matching bipartite approach. Constraints based on LPP or Integer Programming problem is developed for projective sentences. It could produce multiple parse tree after enormous computation to develop constraints and obtain the Maximum parse tree. Illustration is as follows:

ઘનશ્યામ	હાથથી	કેળું	ખાય છે. [Eng: Ghanshyam eats banana with his Hand.]
K1	K3	K2	M.V.
a	b	c	A

Using Constraint rule it gives

$y_1\ (karta) = 1$, $y_4\ (karma) = 1$, $y_5\ (karna) = 1$ (i.e. Ghanshyam eats banana.)

Or

$y_2\ (karta) = 1$, $y_3\ (karma) = 1$, $y_5\ (karna) = 1$ (i.e. Banana eats Ghanshyam.)

[Detailed in section 6.3.3]

The approach of maximum matching bipartite graph is explored to parse the projective sentence of Gujarati language which is computationally less costly as compared to LPP. The delimitation of this approach is it fails to parse the nonprojective sentence. Illustration is as follows: એ છોકરો કે જે ત્યાં બેઠો છે એ મારો ભાઈ છે.(Eng: The Boy who is sitting there is my brother.)It fails this because it not satisfied the maximum matching bipartite rule because in this sentences there are more then one verb..i,e, બેઠો છે and ભાઈ છે.

In order to parse nonprojective, multiverbal sentences the novel approach based on Conceptual Graph Assembly is developed for intra-chunk level parser. For that purpose, NL-support and revised libelling techniques are introduced. Number of projective and non-projective sentences are being parsed using Conceptual graph-based parser manually.

Part of Speech Tagging has always been a challenging task in the era of Natural Language Processing. This article presents POS tagging for Gujrati text using Hidden Markov Model and Viterbi algorithm. From the Gujrati text annotated corpus training and testing data set are randomly separated. Both the methods are employed on the data sets. Viterbi algorithm is found to be computationally faster and accurate as compared to HMM. Error analysis where the mismatches took place is elaborately discussed. SVM Based technique for Tagging and its comparison with Viterbi, it discussed in section [6.2] Since SVM was taking longer time to converge, wavelet-based feature extraction technique is employed to reduce the complexity of the SVM tagger.

Syntactic parsing is an important undertaking which is required for NLP applications including machine interpretation. It is a testing assignment to build up a subjective parser for morphological rich and agglutinative dialects. Syntactic investigation is utilized to comprehend the linguistic structure of a characteristic dialect sentence. It yields all the linguistic data of each word and its constituent. Likewise, issues identified with it assist us with understanding the dialect in a more point by point way. This writing study is preparation to comprehend the distinctive parser advancement for Indian dialects and different methodologies that are utilized to grow such apparatuses and procedures. These references give a study of research papers from surely understood diaries and meetings.

2. Brief description on the state of the art of the research topic:

Our main task of thesis is to built a system which can annotate part-of-speech for Gujarati texts automatically, with the help of various machine learning algorithms. BIS (Bureau of Indian standards) Part of Speech Tag Set for Gujarati is used to develop the part of speech tagger. Currently in trend the Bidirectional LSTM and Condition random filed used in different languages, but we explore viterbi algorithm, Support vector machine, Artificial Neural network and wavelet for Gujrati languages.

3. **Definition of the problem**:

A large number of problems in the field of NLP for Gujarati language are not much explored. Depending up on the literature review on it. Context free Parsing rules as a part of morphological analysis for Gujarati language. The concepts of dynamic programming problem, Graph theory, fuzzy logic and Artificial neural networks need to be explored for this purpose. There are seldom research articles available in the literature for Techniques based on Image compression and approximation (in the case of broken glyphs) using multiwavelets. I propose to work on such techniques for Gujarati grammar using bipartite graph, conceptual graphs approach.

4. Objective and Scope of work

Morphological analysis of the grammar of Gujarati language is not much explored. The objective is to develop a context free parser for Gujarati languages. To general POS tagger and consequently grammar-based Gujarati parser so as that can be used for machine translation and lexical analysis. Future Scope of the work is to enhance the database significantly and develop intra-chunk level parser.

5. Original Contribution by the thesis:

Seldom articles are available in the literature based on NLP as far as Gujarati language grammar is concern [Chirag Patel et al]. The machine learning part is performed using a CRF model. The algorithm has achieved an accuracy of 92% for Gujarati texts where the training corpus is of 10,000 words and the test corpus is of 5,000 words. Viterbi base POS tagger is developed in MATLAB with 92.87% accuracy which is given better accuracy than CRF model. Wavelet based SVM model is developed for POS tagging to increase the speed tagger. The Conceptual graph based parsing technique is introduce in order to develop deep parser (Intra-Chunk parser) in which new definitions like NL-support and labelling are introduced.

6. Methodology of Research, Result/Comparison:

A Machine learning algorithm for Gujarati Part of Speech Tagging has been used by Chirag Patel and Karthik Gali [35]. The machine learning part is performed using a CRF model. The algorithm has achieved an accuracy of 92% for Gujarati texts where the training corpus is of 10,000 words and the test corpus is of 5,000 words. From the experiments they observed that if the language specific rules can be formulated in to features for CRF then the accuracy can be reached to very high extents.

6.1 HIDDEN MARKOV MODEL (HMM)

It calculates the probability of a given sequence of tags. By calculating the probability, it specifies the most suitable tag for a word or token of a sentence that it occurs with the n previous tags, where the value of n is set to 1, 2 or 3 for practical purposes. The most useful algorithm for implementing an n-gram approach is HMM‟s Viterbi Algorithm for tagging new text. HMM is a special case of Bayesian inference. In classification task, we are given some observation and our job is to determine which of a set of classes it belong to. POS tagging is generally treated as a sequence of classification task. So here the observation is a sequence of word(sentences) and it is our job to assign them a sequence of part of speech tags.

6.1.1 VITERBI ALGORITHM

HMM model that contains hidden variables, the task of determining which sequences of variable is the underlying source of some sequence of observation is called the decoding task. The Viterbi algorithm is perhaps the most common decoding algorithm used for HMMs, whether for POS tagging or for speech recognition.

Experimental Procedure for Gujarati text

Total tags: 28

Database: 351 Guajarati words

Size of transition matrix: 28x 28

Size of Emission matrix: 28 x 351

Classification accuracy:

Technique	No of mismatch	Accuracy
Viterbi	25	92.87

6.2 Support Vector Machines Approach:

SVM is a machine learning algorithm has been applied to various practical problems like NLP. For dealing with all the requirements of modern NLP technology the SVM Approach is used because of combining simplicity, flexibility, robustness, portability and efficiency. The support vector machine (SVM) is a universal constructive learning procedure based on the statistical learning theory Vapnik,1995 and Cherkassky and Mulier,1998.

A support vector machine (SVM) is a supervised learning technique from the field of machine learning applicable to both classification and regression. Originally it was worked out for linear two-class classification with margin, where margin means the minimal distance from the separating hyperplane to the closest data points. SVM learning machine seeks an optimal separating hyperplane, where the margin is maximal. An important and unique feature of this approach is that the solution is based only on those data points, which are at the margin. These points are called support vectors. The linear SVM can be extended to nonlinear one when first the problem is transformed into a feature space using a set of nonlinear basis functions. In the feature space - which can be very high dimensional - the data points can be separated linearly. An important advantage of the SVM is that it is not necessary to implement this transformation and to determine the separating hyperplane in the possibly very-high dimensional feature space, instead a kernel representation can be used, where the solution is written as a weighted sum of the values of certain kernel function evaluated at the support vectors.

Support Vector machines (SVM) are a new statistical learning technique that can be seen as a new method for training classifiers based on polynomial functions, radial basis functions, neural networks, splines or other functions. Support Vector machines use a hyper-linear separating plane to create a classifier. For problems that cannot be linearly separated in the input space, this machine offers a possibility to find a solution by making a non-linear transformation of the original input space into a high dimensional feature space, where an optimal separating hyperplane can be found. Those separating planes are optimal, which means that a maximal margin classifier for the training data set can be obtained. Rychetsky (2001). The support vector machine (SVM) is a supervised learning method that generates input-output mapping functions from a set of labelled training data Wang (2005).

Table 1: Training and testing data set using Python by SVM technique

Total data 1373 (Total Tag used 35)	**Total data 1373 (Total Tag used 35)**
Training data 595	Training data: 780
Testing data 778 (Correct 771	Testing data: 593 (Correct 582)
Classification Accuracy 99.10%	Accuracy: 98.14 %

6.3 Dependency Parsing for Guajarati Languages:

6.3.1 Parsing

Parsing is a method of understanding the grammatical structure of the sentences which can be solved with different method like, Linear programming problem and Bipartite matching problem. Parsing is the one of the major tasks which help in understanding the natural languages. It is useful in several natural language application, like machine translation, anaphora resolution etc. In machine translation, It is the process by which computer software is used to translate text from one natural language (such as English) to another (such as Gujarati). Sometimes a machine translate will be wrong output. For example.

1. Marry is Good in nature. (લગ્ન પ્રકૃતિ સારી છે.)
2. The Soldiers were searching for people with Helicopters. (સૈનિકો હેલિકોપ્ટરવાળા લોકો માટે શોધક હતા.)

There is two type of Dependency graph for the sentences:

Projective and Non –Projective

Projective: રામે રાવણ ને તીર થી માર્યો.

Non projective sentence: રામ ફળ ખાઈ ને મોહન ને બોલાવે છે.

6.3.2 Parsing Using Bipartite graph:

Bipartite graph:

In the mathematical field of graph theory, a bipartite graph is a graph whose vertices can be divided into two disjoint and independent sets and such that every edge connects a vertex in to one in. A bipartite graph G = (V_C, V_R, E) where V_C, V_R are the set of nodes such that $V_C \cup V_R$ =ϕ and E is the set of edges between V_C and. V_R.The bipartite graph constraints in three stages.

Matching: A matching of a bipartite graph G=(U,V,E) is a subset of edges with the property that no edges of M share the same node . The matching problem is to find a maximal matching of G , that is , a matching with the largest number of edges.

Maximal Bipartite graph: It is the largest subset of edges in a bipartite graph such that no two selected edges share a common vertex.

Planar Graph: It can be drawn in such a way that no edges cross each other. i.e., A planar graph is a Graph that can be embedded in the plane.

6.3.3 Parsing using LPP for Projective:

A parse can be obtaining from the constraints graph using integer programming. A contains is converted in to an integer programming problem by introducing a variable X for an arc from node i to j labelled by karaka K in the constraints graph such that there is a variable. The variable takes their values as 0 and 1 Equality and inequality constraints in the integer programming problem can be obtain from the condition as below:

1. For each demand group i for each of its mandatory karakas k the following equality must hold $M_{i,k} : \sum_j x_{i,k,j} = 1$
    i. Not that $M_{i,k}$ stands for the equation form a given demand word i and karaka k thus there will be as many equation as combination of i and k
2. For each demand group i , for each of its optional karakas k the following

   inequality must hold $O_{i,k} : \sum_j x_{i,k,j} \leq 1$
3. For each of the source group j, the following equality must hold

$$S_j : \sum_{i,k} x_{i,k,j} = 1$$

thus, their will as many equations as there are source word

1. ઘનશ્યામ હાથથી કેળું ખાય છે.

K1	K3	K2	m.v.
a	b	c	A

Constraint C1: $M_{A,K1} : x_{A,K1,a} + x_{A,K1,c} = 1$ $M_{A,K2} : x_{A,K2,a} + x_{A,K2,c} = 1$

Constraint C2: $O_{A,K3} : x_{A,K3,b} \leq 1$

Constraint C3: $S_a : x_{A,K1,a} + x_{A,K2,a} = 1$

$S_b : x_{A,K3,b} = 1$

$S_c : x_{A,K1,c} + x_{A,K2,c} = 1$

Take $x_{A,K1,a} = y_1$, $x_{A,K1,c} = y_2$, $x_{A,K2,a} = y_3$, $x_{A,K2,c} = y_4$, $x_{A,K3,b} = y_5$

We get $M_{A,K1} : y_1 + y_2 = 1$

$M_{A,K2} : y_3 + y_4 = 1$

$O_{A,K3} : y_5 \leq 1$

$S_a : y_1 + y_3 = 1$, $S_b : y_5 = 1$, $S_c : y_2 + y_4 = 1$

The cost function to be minimized is:

Min Z = $y_1 + y_2 + y_3 + y_4 + y_5$

Subject to , $y_1 + y_2 = 1$

$y_3 + y_4 = 1$

$y_5 \leq 1$

$y_1 + y_3 = 1$

$y_5 = 1$

$y_2 + y_4 = 1$

Result : $y_1 = 1$, $y_4 = 1$, $y_5 = 1$

6.3.4. Parsing using LPP for Non-Projective:

1.માણસે નાવ બનાવવામાટે લાકડી કાપી .

K1	K2	Verb.modi.	K2	M.V.
a	b	B	c	A

Constraint C1: $M_{A,K1} : x_{A,K1,a} = 1$ $M_{A,K2} : x_{A,K2,b} + x_{A,K2,c} = 1$, $M_{B,K2} : x_{B,K2,b} = 1$

Constraint C2: $O_{A,K4}$: 0

Constraint C3: $S_a : x_{A,K1,a} = 1$

$S_b : x_{A,K2,b} + x_{B,K2,b} = 1$

$S_c : x_{A,K2,c} = 1$

Take $x_{A,K1,a} = y_1$, $x_{A,K2,b} = y_2$, $x_{A,K2,c} = y_3$, $x_{B,K2,b} = y_4$

We get $M_{A,K1} : y_1 = 1$

$M_{A,K2} : y_2 + y_3 = 1$

$M_{B,K2} : y_4 = 1$

$S_a : y_1 = 1,$

$S_b : y_2 + y_4 = 1$, $S_c : y_3 = 1$

The cost function to be minimized is:

Min Z = $y_1 + y_2 + y_3 + y_4$

Subject to , $y_1 = 1$

$y_2 + y_3 = 1$

$y_4 = 1$

$y_3 = 1$

$y_2 + y_4 = 1$

Result : $y_1 = 1$, $y_3 = 1$, $y_4 = 1$

Table:6.1 Abbreviation for the Karakas

Abbreviation	Karaka
K1	Karta
K2	Karma
K3	Karana

6.4 Conceptual Graph Based Parsing:

To define a Conceptual Graph for a given sentence (Gujarati) we define the following first:

Definition 1. Ordered bipartite graph: defined as B = (V_C, V_R; E_B, l). Which is formed by first considering a problem bipartite graph G = (V_C, V_R, E) defined in section 2 and then defining a linear ordering ∀ w_i ∈ V_C on set of edges incident to w_i. where V_R represents the verb words paired with all the possible mandatory and optional karakas and V_C is the set of all the non-verb words. An ordered bipartite graph requires that for each node in one of the classes of the bipartition, its neighbours (belonging to the other partition) to be ordered. For every verb-karaka combinations (v_jk) for each non-verb words (w_i), we define a labelling

$l_i: E_B \to \{1,2, \ldots .. |V_R|\}$, on the edges of B by,

$$l_i[v_jk, w_i] \forall k \in T_{R,} \forall w_i \in V_c = \begin{cases} 1, & if\ k\ is\ mandatory\ and\ w_i\ is\ NNP\ or\ NN\ \forall j \\ 2 & if\ K\ is\ Optinal\ \ and\ w_i\ is\ NN\ \forall j \\ 3, & Other\ wise \end{cases}$$

Where i= 1,2,…..$|V_C|, j = 1,2 \ldots . |V_R|$and $l_i(\{v_jk, w_i\})$ is the index of the edge $(\{v_jk, w_i\}$ in the above ordering of the edges incident in B to V_C. The label $l_i \forall i \in |V_C|$, is called the order labelling of the edges of B. Now we have for each $v_jk \in V_R, N_B^p(v_jk)$denotes the p – th neighbour of v_jk, i.e,$w_p = N_B^p(v_jk)$ iff $(\{v_jk, w_p\} \in E_B$. Given a node$x_m \in V_C \cup V_R$, $\overline{N_B}\ (x_m)$ denotes the neighbours set of this node, i.e.$\overline{N_B}\ (x_m) = \{x_m \in\ V_C \cup V_R / \{x_m, u_m\} \in E_B\}$. Similarly, if ⊆ $V_C \cup V_R$, , its neighbours set is denoted as , $\overline{N_B}\ (x_m) = \cup_{x_m \in A} \overline{N_B}\ (x_m) - A$

We further assume that for each $w_i \in V_C$ there is $v_jk \in V_R$ and n ∈ N such that $w_i = N_B^n(v_jk)$ i.e. B has no isolated vertices. Ordered bipartite graphs are appropriate tools to represent and visualize (directed) hypergraphs. Visually, an ordered bipartite graph can be represented using boxes for vertices in VC, ovals for vertices in VR and integer labelled simple curves (edges) connecting boxes and ovals.

Definition 2: NL-Support : NLS=($T_C, T_R, I,*$) is a NL Support related with the syntax and semantics of the natural language where:

- T_C is a finite, partially ordered set odf concept types
- $T_R$is a finite poset of relation types of arity 2 ($T_R{}^2$) and T_R $*$ (set of generic relations),
- I is the set of countable set of individual markers, used to refer specific concepts and
- $*$ is the generic marker used to refer to an unspecified concept (having, however, a specified type).

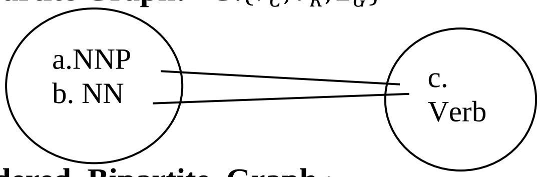
Implementation for Projective sentences:

(1) બાળક કેળું ખાય છે. [ENG The Child eating the Banana.]

NN NNP M.V.

BG: Verb: ખાય છે (Eating) V_C : $\{NNP, NN\}$ V_R : $\{Verb\}$ e_1: $\{a, c\}$$e_2$: $\{b, c\}$

Bipartite Graph: G:$\{V_C, V_R, E_G\}$



Ordered Bipartite Graph :

$N_G : V_R \rightarrow V_C^+ \ \ is\ a\ Mapping$

N_G(Verb) :$\{NNP, NN\}$

$r \in V_R$,$N_G^i(r)$: $i^{th}\ neighbour\ of\ r.$

i .e ,$v = N_G^i(r), iff\ \{r, v\} \in\ E_G\&\ \ l(r, v) = i$

i,e$l(e_1) = 1$,$l(e_2) = 2$

NL-Support :

T_C: Specific structure, such as a tree, a lattice or a semi lattice .

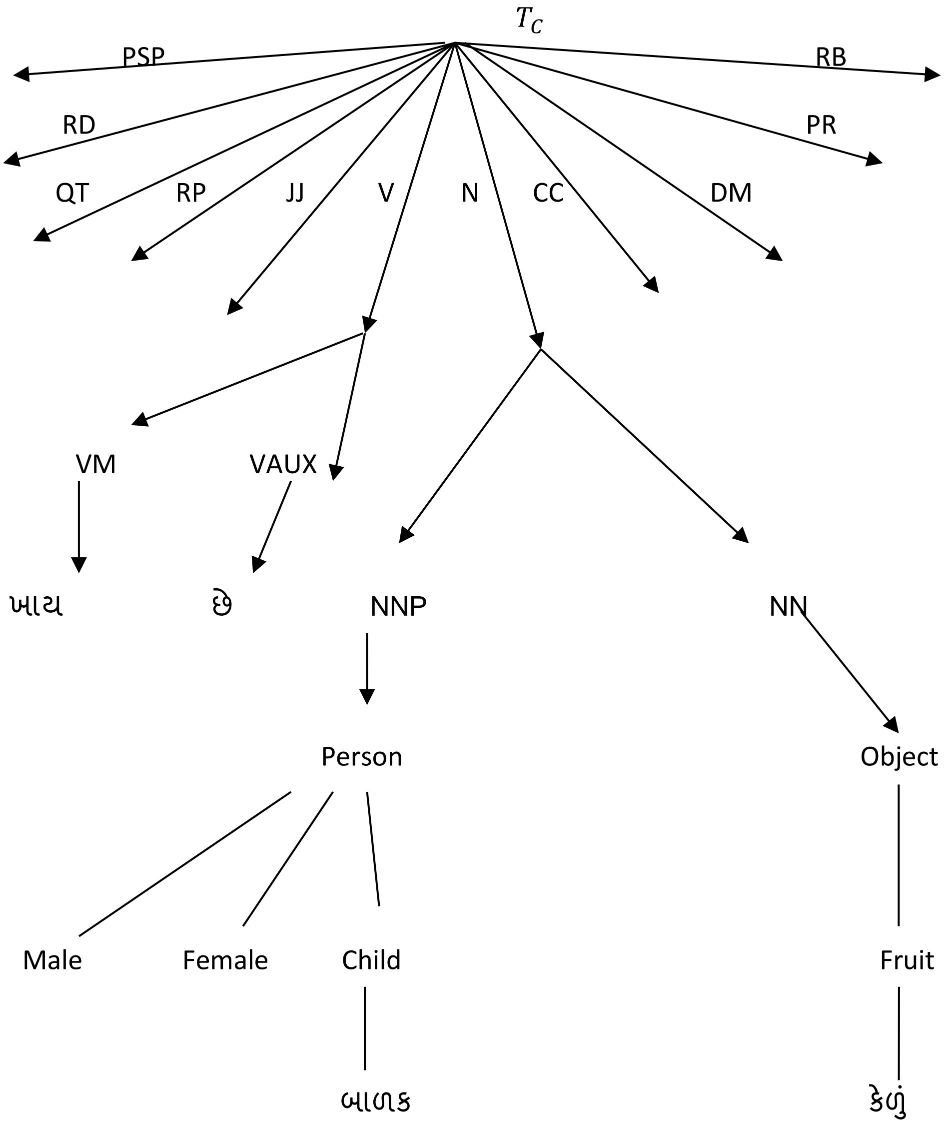
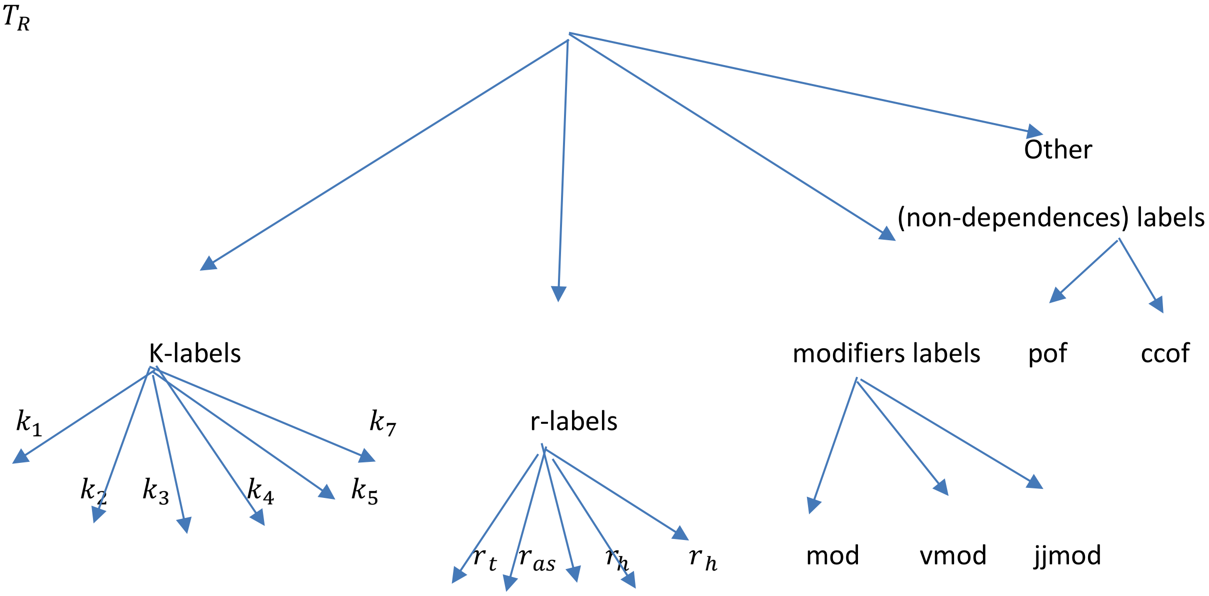


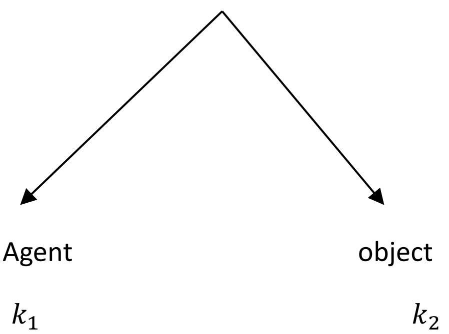
Fig 6.2 Concept type Hierarchy

T_R : relation type partially ordered set $\leq$.

T_R



T_R^2 :



i.e. $(T_R^i, \leq)$ $i = 1,2$

Fig 6.3 Relation type Hierarchy

Labeling :ʎ $: G \rightarrow S$ $\quad r \in V_R \quad i, e. Verb \in V_R$ ʎ(verb) $\in T_R^2$, ʎ(verb) = type(r) =ખાયછે.

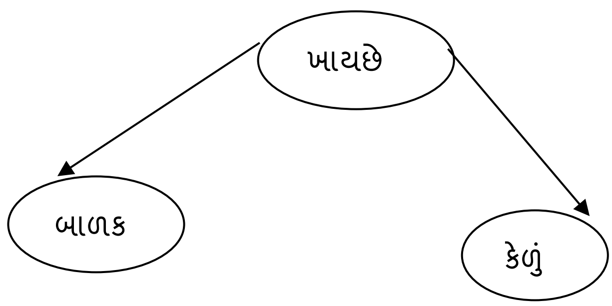
a ∈ V_Cʎ(a) ∈બાળક×(∗)

b∈ V_Cʎ(b) ∈કેળું×(I)

Verb Frame :(ખાયછે) Ref [5]

Arc-label	Rule	Necessity	vibhakti	Lexical type
k_1	Ergative	M	0	n
k_2	Accusative	M	0	n

Fig: 1 is the Desired Solution.



7. Achievements with respect to the objectives:

We use different technique in Guajarati pos tag data set. And for the parsing of Gujarti sentences we describe the new definition of NL- support and ordered Bipartite graph. It is very useful for the researcher to develop another thing for the Gujrati languages.

8. Conclusion:

- Investigations delineated in four tables in segment, viterbi performs reliably better in all the four classes when contrasted with SVM. As highlighted in figure 2, In 56 cases SVM does not label accentuation stamps legitimately. In the greater part of the cases it is misclassified with VM TAG. Anyway, that isn't the situation for Viterbi.
- Conceptual graphs represent knowledge as a set of factual assertions. This is useful for representing a purely deterministic world, but there are some cases in which this is not expressive enough.
- In these cases, we would like to be able to express scenarios, alternatives, interdependent events.
- A syntactical and semantical mechanism was introduced to capture these requirements in a simple way, consistent with the conceptual graph's representation spirit.
- Viterbi algorithm is found to be very accurate as compared to HMM for POS Tagging of Guajarati text. The frequency of misclassification of the Tag DMD is 6. The second highest frequency of misclassified Tag VM&VAUX is 4.

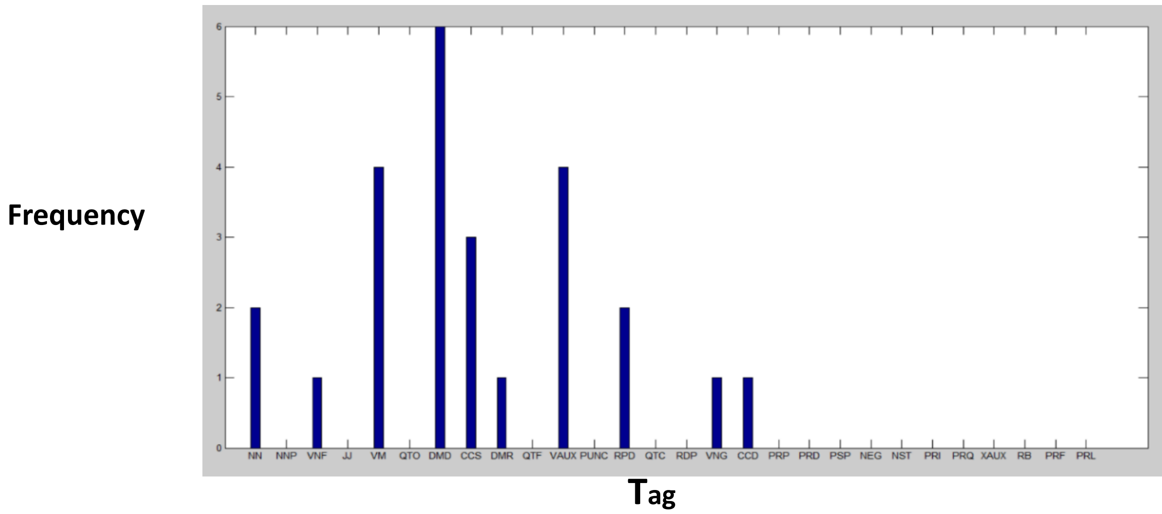


Fig: 2 Error analysis table

- We define a CG model to the parsing of two languages and which further can be implemented for any natural language with respect to the context of the syntax and semantics of the language. Using CG approach for parsing of a natural language have many advantages over other existing methods, the existence of a NL support distinguishes and sorts the hierarchy of concepts and relation. We re-define ordered bipartite graph, NL support in a new context of parsing. The derived mathematical properties could assist future work in research and the development of knowledge representation, in particular, in the area of parsing, which has many applications in Natural Language Processing. We finally propose a theorem to construct a CG as a parse of a non-projective sentence by decomposing it into-sub parts (projective).

9. Copies of paper published and a list of all publication arising from the thesis:

[1] M. prajapati Archit Yajnik, "Part of Speech Tagging Using Statistical Approach for Gujrati Text," *Int. J. Appl. Res. Sci. Eng.*, vol. 11, no. 1, pp. 76–79, 2017.

[2] A. Y. Manisha Prajapati, "POS Tagging of Gujarati Text using VITERBI and SVM," *Int. J. Comput. Appl.*, vol. 181, no. 43, pp. 32–35, 2019.

[3] Manisha Prajapati and Archit Yajnik, "Parsing for Indian Languages A Literature Survey," *Int. J. Comput. Sci. Eng.*, vol. 6, no. 8, pp. 1009–1018, 2018.

9.1. Copies of paper communicate and a list of all publication arising from the thesis:

[1] M. Prajapati Archit Yajnik, "Constraint Based Gujarati Parser Using LPP,"*IC4S-2019*.

Publication by Springer lecture notes in Networks and Systems.

[2] A. Yajnik and Manisha prajapati, "A Conceptual Graph Approach to the Parsing of Projective Sentences," *Int. Journal of Mathematics and Computer Science*.

10. References

[1] Ȧ M. T. M. and Ḃ D. C. V., "Survey : Natural Language Parsing For Indian Languages," 2015.

[2] R. Abstract, P. O. Speech, N. L. Processing, E. Schedule, I. Constitution, and I. Constitution, "Parts Of Speech Tagger for Maithili Language Using HMM," vol. 7, no. 4, pp. 206–211, 2018.

[3] P. J. Antony and K. P. Soman, "Ke[1] P. J. Antony and K. P. Soman, 'Kernel based part of speech tagger for Kannada,' 2010 Int. Conf. Mach. Learn. Cybern. ICMLC 2010, vol. 4, no. July, pp. 2139–2144, 2010.rnel based part of speech tagger for Kannada," *2010 Int. Conf. Mach. Learn. Cybern. ICMLC 2010*, vol. 4, no. July, pp. 2139–2144, 2010.

[4] M. prajapati Archit Yajnik, "Part of Speech Tagging Using Statistical Approach for Gujrati Text," *Int. J. Appl. Res. Sci. Eng.*, vol. 11, no. 1, pp. 76–79, 2017.

[5] R. Begum, S. Husain, L. Bai, and D. M. Sharma, "Developing Verb Frames for Hindi," pp. 1925–1932, 2003.

[6] R. Begum and D. M. Sharma, "Development and Analysis of Verb Frame Lexicon for Hindi," vol. 5, no. 1, pp. 1–22, 2017.

[7] A. Bharati, S. Husain, B. Ambati, S. Jain, D. Sharma, and R. Sangal, "Two semantic features make all the difference in parsing accuracy," *Proc. 6th Int. Conf. Nat. Lang. Process.*, vol. 8, no. November, 2008.

[8] A. Bharati, M. Bhatia, V. Chaitanya, and R. Sangal, "Paninian Grammar Framework Applied to English," pp. 1–23, 1996.

[9] A. Bharati, M. Gupta, V. Yadav, K. Gali, and D. M. Sharma, "Simple parser for Indian languages in a dependency framework," *Proc. Third Linguist. Annot. Work.*, no. August, pp. 162–165, 2009.

[10] S. Chatterji and D. Roy, "Grammar Driven Rules for Hybrid Bengali Dependency Parsing," 2009.

[11] W. Chen, Z. Li, and M. Zhang, "Dependency Parsing : Past , Present , and Future."

[12] N. Choudhary, "Human Language Technology Challenges for Computer Science and Linguistics - 5th Language and Technology Conference, LTC 2011, Revised Selected Papers," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8387 LNAI, no. February 2015, 2014.

[13] M. Croitoru, "Conceptual Graphs at Work : Efficient Reasoning and Applications," *Science (80-.).*, 2006.

[14] M. Croitoru, "A Conceptual Graph Approach to the Generation of Referring Expressions," pp. 2456–2461, 1984.

[15] M. Croitoru and E. Compatangelo, "A tree decomposition algorithm for Conceptual Graph projection," *Proceedings, 10th Int. Conf. Princ. Knowl. Represent. Reason. KR 2006*, pp. 271–276, 2006.

[16] M. Croitoru and E. Compatangelo, "Conceptual Graph Assemblies digs," 2002.

[17] M. Croitoru, B. Hu, S. Dashmapatra, P. Lewis, D. Dupplaw, and L. Xiao, "A Conceptual Graph Based Approach to Ontology Similarity Measure," *Lnai*, vol. 4604, no. di, pp. 154–164, 2007.

[18] B. R. Das, S. Sahoo, C. S. Panda, and S. Patnaik, "Part of speech tagging in odia using support vector machine," *Procedia Comput. Sci.*, vol. 48, no. C, pp. 507–512, 2015.

[19] J. Dholakia, “Wavelet Feature Based Confusion Character Sets for Guja[1] J. Dholakia, ‘Wavelet Feature Based Confusion Character Sets for Gujarati Script,’ pp. 371–376, 2007.rati Script,” pp. 371–376, 2007.

[20] P. S. Dholakia and M. M. Yoonus, “Rule Based Approach for the Transition of Tagsets to Build the {POS} Annotated Corpus,” *Int. {J}ournal {A}dvanced {R}esearch {C}omputer {C}ommunication {E}ngineering*, vol. 3, no. 7, pp. 7417–7422, 2014.

[21] A. Ekbal and S. Bandyopadhyay, “Part of speech tagging in Bengali using Support vector Machine,” *Proc. - 11th Int. Conf. Inf. Technol. ICIT 2008*, pp. 106–111, 2008.

[22] M. M. El Mostapha Chakir, Chancerel Codjovi, Youness Idrissi Khamlichi, “International Journal of Advanced Research in Computer Science and Software Engineering,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 5, no. 4, pp. 77–85, 2015.

[23] A. K. Eragani and V. Kuchibhotla, “Improving malt dependency parser using a simple grammar-driven unlexicalised dependency parser,” *Proc. Int. Conf. Asian Lang. Process. 2014, IALP 2014*, pp. 211–214, 2014.

[24] F. Gavril, “Generating the maximum spanning trees of a weighted graph,” *J. Algorithms*, vol. 8, no. 4, pp. 592–597, 1987.

[25] J. Hall and J. Nilsson, “MaltParser : A Data-Driven Parser-Generator for Dependency Parsing,” pp. 2216–2219.

[26] K. Hall, “K-best spanning tree parsing,” *Proc. 45th Annu. Meet. Assoc. Comput. Linguist.*, vol. 45, no. May, pp. 392–399, 2007.

[27] S. Haykin, *Neural Networks and Learning Machines*, vol. 3. 2008.

[28] L. S. Hng, “Lee.1990.pdf.” .

[29] S. Journal and I. Factor, “International Journal of Advance Engineering and Research Development,” pp. 464–467, 2015.

[30] P. Kosaraju, S. R. Kesidi, V. Bhargav, and R. Ainavolu, “Experiments on Indian Language Dependency Parsing,” 1995.

[31] B. V. S. Kumari and R. R. Rao, “Hindi Dependency Parsing using a combined model of Malt and MST,” no. December, pp. 171–178, 2012.

[32] K. Nongmeikapam, “Manipuri Chunking : An Incremental Model with POS and RMWE,” no. December, pp. 277–286, 2014.

[33] K. Nongmeikapam and S. Bandyopadhyay, “Genetic Algorithm (GA) Implementation for Feature Selection in Manipuri POS Tagging,” *Proc. 13th Int. Conf. Nat. Lang. Process.*, no. December, pp. 267–274, 2016.

[34] G. D. Parsing, “Marina Valeeva.”

[35] C. Patel and K. Gali, “Part-Of-Speech Tagging for {G}ujarati Using Conditional Random Fields,” *Proc. IJCNLP-08 Work. NLP Less Privil. Lang.*, no. January, pp. 117–122, 2008.

[36] S. Petrov, “Syntax and Parsing,” *Comput. Linguist.*, no. 1979, pp. 577–578, 1995.

[37] B. Plank, “Natural Language Processing : Introduction to Syntactic Parsing,” 2012.

[38] and U. G. Sankar De, Arnab Dhar, “ICON09 NLP TOOLS CONTEST : INDIAN LANGUAGE DEPENDENCY PARSING Hyderabad , India,” no. December, 2009.

[39] A. Secer, A. C. Sonmez, and H. Aydin, “Ontology mapping using bipartite graph,” *Int. J. Phys. Sci.*, vol. 6, no. 17, pp. 4224–4244, 2011.

[40] T. B. Shahi, “Support Vector Machines based Part of Speech Tagging for Nepali Text,”

vol. 70, no. 24, pp. 38–42, 2013.

[41] N. Shimizu, “Maximum Spanning Tree Algorithm for Non-projective Labeled Dependency Parsing,” *Comput. Linguist.*, no. June, pp. 236–240, 2006.

[42] M. T. Makwana and D. C. Vegda, “Survey:Natural Language Parsing For Indian Languages.”

[43] K. Singla, A. Tammewar, N. Jain, and S. Jain, “Two-stage Approach for Hindi Dependency Parsing Using MaltParser,” no. December, pp. 163–170, 2012.

[44] J. F. Sowa, “Conceptual Graphs,” *Found. Artif. Intell.*, vol. 3, no. Findler 1979, p. 213−237, 2008.

[45] Y. Tajiri, R. Yabuwaki, T. Kitamura, and S. Abe, “Machines,” pp. 108–109, 2010.

[46] L. Technologies, “Dependency parsing approaches for Indian Languages : Hindi and Sanskrit,” no. July, 2014.

[47] O. This, I. Bgs, S. C. Graphs, and N. Bgs, “Basic Conceptual Graphs.”

[48] P. Twardowski *et al.*, “A phase i trial of mushroom powder in patients with biochemically recurrent prostate cancer: Roles of cytokines and myeloid-derived suppressor cells for Agaricus bisporus-induced prostate-specific antigen responses,” *Cancer*, vol. 121, no. 17, pp. 2942–2950, 2015.

[49] A. Vaidya, “A classification of dependency labels in the Hindi Treebank Hindi / Urdu Treebank.”

[50] A. Y. Manisha Prajapati, “POS Tagging of Gujarati Text using VITERBI and SVM,” *Int. J. Comput. Appl.*, vol. 181, no. 43, pp. 32–35, 2019.

[51] manisha prajapati and Archit Yajnik, “Parsing for Indian Languages A Literature Survey,” *Int. J. Comput. Sci. Eng.*, vol. 6, no. 8, pp. 1009–1018, 2018.

[52] G. Mcdonald and C. Macdonald, “A Study of SVM Kernel Functions for Sensitivity Classification Ensembles with POS Sequences,” no. June, pp. 7–11, 2017.

[53] A. Mukherjee, S. Kübler, and M. Scheutz, “POS Tagging Experts via Topic Modeling,” *Proc. 13th Int. Conf. Nat. Lang. Process.*, pp. 120–128, 2016.

[54] A. Nietzio, “Support Vector Machines for Part-of-Speech Tagging,” pp. 5–8, 2002.

[55] J. Nivre, “Parsing Indian Languages with MaltParser,” *Proc. ICON09 NLP Tools Contest Indian Lang. Depend. Parsing*, no. January, pp. 12–18, 2009.

[56] J. Nivre *et al.*, *MaltParser: A language-independent system for data-driven dependency parsing*, vol. 13, no. 2. 2007.

[57] A. Vaidya, J. D. Choi, M. Palmer, and B. Narasimhan, “Analysis of the Hindi Proposition Bank using Dependency Structure Analysis of the Hindi Proposition Bank using Dependency Structure,” no. May 2018, 2011.

[58] B. Venkata, S. Kumari, and R. R. Rao, “Improving Indian Language Dependency Parsing by Combining Transition-based and Graph-based Parsers,” *Int. J. Comput. Appl.*, vol. 115, no. 5, pp. 975–8887, 2015.

[59] A. Yajnik, “Part of Speech Tagging Using Statistical Approach for Nepali Text,” vol. 11, no. 1, pp. 76–79, 2017.

[60] S. A. Yasuyuki Tajiri, Ryosuke Yabuwaki, Takuya Kitamura, “Feature extrction using SVM.” ICON NIP, pp. 1–8, 2010.