Verbframator: Semi-Automatic Verb Frame Annotator Tool with Special Reference to Marathi

Hanumant Redkar, Sandhya Singh, Nandini Ghag, Jai Paranjape, Nilesh Joshi,

Malhar Kulkarni, Pushpak Bhattacharyya

Center for Indian Language Technology,
IIT Bombay, Mumbai, India
{hanumantredkar,sandhya.singh,samskritkanya,jai.para20,joshinilesh60}@gmail.com,
malhar@iitb.ac.in, pb@cse.iitb.ac.in

Abstract

The sentence is incomplete without a verb in a language. A verb is majorly responsible for giving the meaning to a sentence. Any sentence can be represented in the form of a verb frame. Verb frames are mainly developed as a knowledge resource which can be used in various semantic level Natural Language Processing (NLP) activities. This paper presents the *Verbframator* – a verb frame annotator tool which automatically extracts and generates verb frames of example sentences from Marathi wordnet. It also helps in generating Shakti Standard Format (SSF) files of the given example sentences. The generated verb frames and SSF files can be used in the dependency tree banking and other NLP applications like machine translation, paraphrasing, natural language generation, *etc*.

1 Introduction

Verb is one of the most important part-of-speech (POS) category in any language. It plays a major role in understanding or defining a sentence. Sentences in any language can be represented in the form of verb frames. *Verb frame* is made of verbal propositions and arguments of words around a given verb. It is required for creating dependency relations which leads to the development of a knowledge resource. These kinds of resources are always required for semantics related Natural Language Processing (NLP) applications like machine translation, natural language generation, paraphrasing, *etc.* Development and analysis of verb frames for Indian languages is an emerging research topic in the field of NLP and computational linguistics.

This paper presents a verb frame generator tool to semi-automatically annotate and generate verb frames using example sentences from wordnet. Marathi wordnet is used as a reference wordnet for this purpose. Marathi[1] is an Indo-Aryan language spoken prominently in Maharashtra, India. Marathi wordnet[2] comes under the umbrella of IndoWordNet[3] which currently has around 32183 synsets and 43834 words, of which 3047 are verbs. The objective is to generate verb frames for these verbs using examples from Marathi wordnet.

The paper covers explanation of the concept, working of verb-frame tool, advantages and disadvantages of the tool and its applications. The immediate application of the output of this tool is to study Marathi sentences and their structures with the help of its generated verb-frames. This study will lead to development of rules and guidelines for verb-frame annotation process.

In computational linguistics and natural language processing, only few efforts have been taken to create this kind of resource for Indian languages. And in spite of Marathi being a morphologically rich, with sufficient resource availability, no remarkable work has been found.

In this paper, section 2 briefs the related work. Section 3 defines the verb frame. Section 4 discusses the work flow of verb frame annotator tool. Section 5 lists features and limitations of the tool. Section 6 concludes the paper with scope for improvement in the tool, followed by references.

[1] https://en.wikipedia.org/wiki/Marathi_language
[2] http://www.cfilt.iitb.ac.in/wordnet/webmwn/
[3] http://www.cfilt.iitb.ac.in/indowordnet/

D S Sharma, R Sangal and A K Singh. Proc. of the 13th Intl. Conference on Natural Language Processing, pages 299–304,
Varanasi, India. December 2016.

2 Related Work

Verb POS easily represents the activity, state or action in a text which can lead to its semantic interpretation. This property of verb is useful in creating the necessary knowledge base required for NLP applications. Though, for English language, there is a substantial amount of work which has been done regarding the development and analysis using verb frames, but only limited work can be seen for Indian languages. This can be seen in the following research work.

Begum et al. (2008) discussed their experience with the creation of verb frames for Hindi language. These frames are further classified based on Paninian grammar framework using 6 karaka relations. The method followed by the authors considers the morphology, syntactic variations and semantics of the verb to divide it into different classes. This resource focuses on NLP related activities for Hindi language.

Based on similar approach, Ghosh (2014) created a resource for verb frames for compound verbs in Bengali language. The objective of the paper was to investigate if the vector verb from the compound verb is able to retain its case marking properties and argument structure or not. And also the knowledge and syntax associated with verb frames can be utilized for categorizing and analyzing the verb words for various NLP applications.

Soni et al. (2013) explored the application of verb frames and the conjuncts in sentence simplification for Hindi language. The method proposed by the authors includes usage of conjuncts as a first level of sentence simplification. This is followed by using verb frames enhanced with tense, aspect and modality features. It is a rule based system and its output was evaluated manually and automatically using BLEU score for the ease of readability and simplification.

Other related work by Schulte (2009), Theakston et al. (2015) has also explored verb frames for English language.

After reviewing these papers, the need for such kind of resource for Indian languages and a tool which can assist in the creation of this type of resources became more apparent. Also, apart from English, language specific verb-frame analysis has been done for few Indian languages such as Hindi and Bengali. However, no attempt has been made for Marathi language in the domain of verb-frame creation and analysis. This paper aims to fill the gap in the literature by providing verb-frame tool for the study of verb frames.

3 What is a Verb Frame?

In any language, verb is a major part-of-speech category. Verbs are used to define actions, activities and states. The potential of verbs to choose their complements (arguments/adjuncts) is referred to as 'verb sub-categorization' or 'verb valency' and a combination of functional components that are evoked by a verb is called as *verb frames*.

Verb frame typically is made of verbal propositions and arguments of words around a verb in a given sentence. Each of the propositional word in a verb frame has arguments such as an arc-label or semantic role label, its necessity in a frame, *vibhakti* (i.e. case marker), lexical type, relation with head verb, position with respect to head verb, *etc*. These verb frames are developed to generate dependency tree structures in a given language. Figure 1 shows a typical verb frame for a Marathi word घड (*ghaDa*, to happen).

Verb::घड
SID::
Verb Sense::
Eng_Gloss::
Verb Class::
Verb_in_Same_Class::
TAM for the verb root::ले
Frames::

Example::त्यांच्या देवघरात समर्थांनी उद्धवस्वामींना दिलेली मिशीवाल्या रामाची दुर्मिळ , ऐतिहासिक मूर्ती , अनोखे श्री यंत्र आणि अशाच काही दुर्मिळ मूर्तींचं दर्शन घडलं .
Frequency::1

FRAME_ID::1

arc_label	necessity	Vibhakti	LexType	posn	reln
k2	m	0	n	1	c
k7p	o	0	n	1	c

Figure 1: Typical verb frame for a word घड

4 Verb Frame Annotator : *Verbframator*

Verb frame annotator interface or *Verbframator* is an online tool developed to extract the verb frames of example sentences from wordnet. It also generates the Shakti Standard Format (SSF) (Bharati et al., 2007) files for these example sentences.

The *Verbframator* is a semi-automatic tool which is developed in PHP and data is stored and maintained in MySQL *verbframe* database. The user input interface of *Verbframator* can be seen in figure 2.

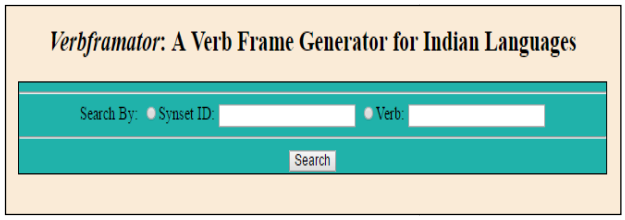

Figure 2: A *Verbframator* user input interface

4.1 Work Flow of *Verbframator*

The *Verbframator* is designed to generate verb frames of the example sentences from wordnet. In this study we have used Marathi wordnet. The work flow of this tool is shown in figure 3. Following are the steps used by *Verbframator* to generate verb frames:

(1) Take wordnet data as an input - synset id or verb:
The *Verbframator* starts with the user input in the form of either verb's synset id or a verb itself. The system is uses example sentences from wordnet for the verb frame annotation process. As shown in figure 2 above, search can be done by entering synset id or a verb in Marathi language.

(2) Extract example sentence(s):

(a) If the user enters verb synset id then system checks if the verb is present in the context of the example sentence and extract the corresponding example sentence.

(b) If the user enters verb itself there are two possibilities: (i) if there is only one synset for the input verb then the system extracts examples of that synset and (ii) if there are more than one synset then the system extracts all the synsets of the input verb. Here user selects synsets one by one and each synset is processed at a time.

In both the cases (a) and (b) above, the examples are extracted only if verb is present in the example sentence, else it will show corresponding error message. Also, if synsets have more than one example sentence then the system extracts only those examples in which the input verb appears.

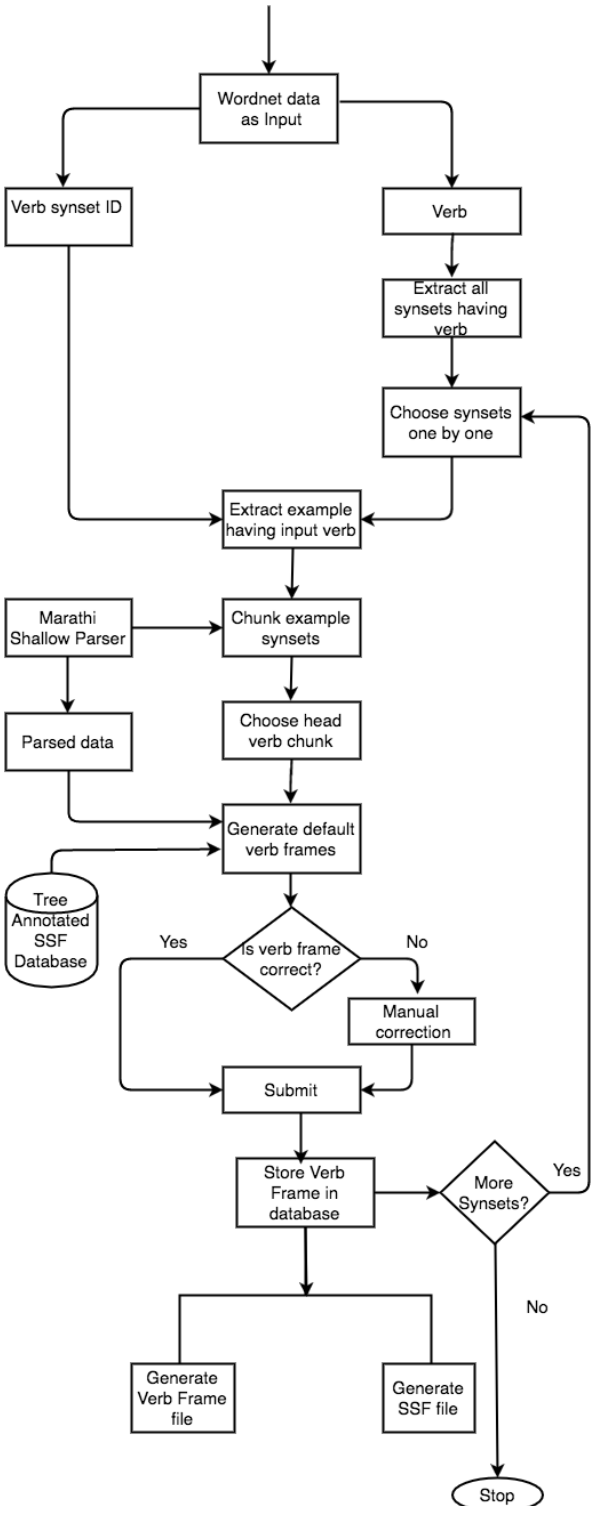

Figure 3: Work flow diagram of *Verbframator*

(3) Chunk example sentence(s):
The input example sentence processed through the Marathi shallow parser[4] which gives the chunked phrases of the input sentence. This chunked output is manually validated by a lexicographer. This can be seen in figure 4. Once the chunking is done the annotator decides and marks the head verb chunk and clicks on 'Extract Verb Frames' button to extract verb frames.

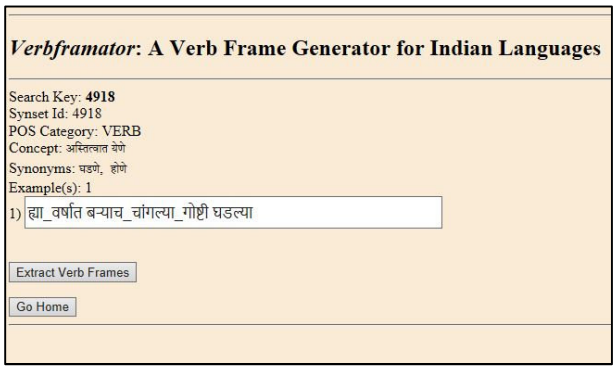

Figure 4: *Verbframator* sentence chunked output

(4) Generate default verb frame(s):
Once the chunking is done, the system automatically generates the default verb frame for an input example.

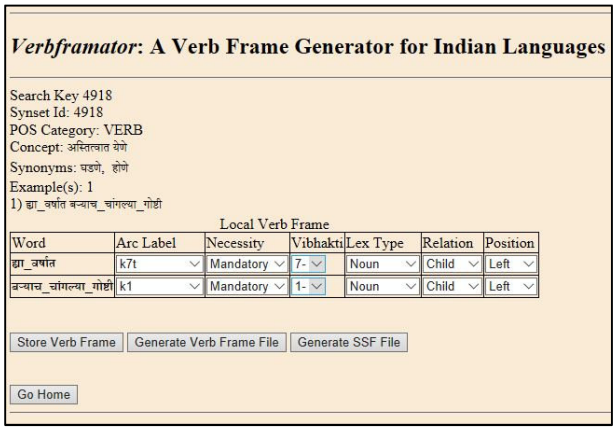

Figure 5: *Verbframator* - default frame generator

[4]http://ltrc.iiit.ac.in/showfile.php?filename=downloads/shallow_parser.php

The method to automatically generate the default verb frames is explained in detail in the next section. The default verb frame generated for a verb घडणे (*ghaDaNe*, to happen, to occur) can be seen in figure 5.

(5) Validate extracted verb frame(s):
The annotator manually validates the generated verb frame. Here, user has to do minimal validation as most of the fields in the verb frame are already entered by the system automatically. Once the validation is done by an annotator, user can store the verb frame, generate the verb frame or/and generate the SSF file.

(6) Store the verb frame(s) data:
Verb frames are stored in the *verbframe* database by using button 'Store Verb Frames'.

(7) Generate verb frame(s):
We can generate the verb frames using button 'Generate Verb Frames'. The verb frame(s) of the processed verb frame is generated in the standard verb frame format. The extracted verb frame can be seen in figure 6.

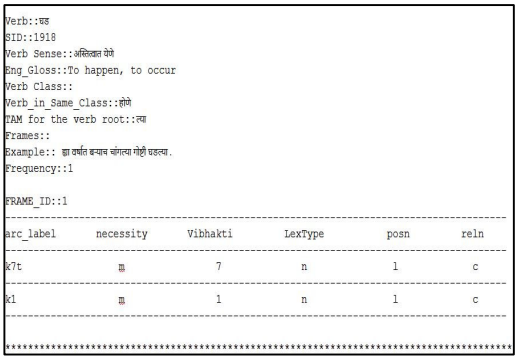

Figure 6: Extracted verb frame for a verb घड

(8) Generate SSF file:
We can also generate SSF file of the processed example sentence by clicking on 'Generate SSF File'.

The process is repeated if there are more synsets for an input verb.

4.2 Method to Generate Default Verb Frames

The default verb frames are automatically generated using (i) the tree annotated data and (ii) Marathi shallow parser output. While extracting verb frame features the *Verbframator* first checks if chunks are already present in the SSF tree annotated database. If chunks are present then the corresponding arc label is extracted from the database else the arc label is extracted from the shallow parser output data.

In extract features using *tree annotated data approach*, we automatically extract verb frame arguments from the tree annotated files. These files are already annotated by the human experts and are available in SSF format. Currently, the system has used about 92 tree annotated SSF files having 5536 sentences, 74476 words and 50813 chunks. This data continuously changes and new annotated data is appended to the existing data. Our offline system preprocesses this tree annotated data and features like arc labels, lex type and other necessary information are extracted from this annotated data. The SSF tree annotated data and extracted features are stored in the *verbframe* database. This is an ongoing process where the feature extraction is done on tree annotated data as and when new tree annotated data is added to the offline preprocessing system.

In extract features using *Marathi shallow parser approach*, if the required features are not available in *verbframe* database then the Marathi shallow parser tool is used to extract these features. The verb frame features such as *vibhakti's* and lexical type can be extracted using the output of the shallow parser tool. The extracted output can be mapped to get the *vibhakti's*. This can be seen in table 1, 2, and 3.

Marathi *Vibhakti* Table					
Sanskrit	**English**	**Sanskrit**	**English**	**Singular Suffixes**	**Plural Suffixes**
Ordinal Number	**Ordinal Number**	**Case Description**	**Case Description**	**(एकवचन)**	**(अनेकवचन)**
prathamā (प्रथमा)	First	kartā (कर्ता)	Nominative case	–	-ā (आ)
dwitīyā (द्वितीया)	Second	karma (कर्म)	Accusative case	-sa (-स), -lā (-ला), -te (-ते)	-sa (-स), -lā (-ला), -nā (ना), -te (-ते)
trutīyā (तृतीय)	Third	karaṇa(करण)	Instrumental case	-nī (नी), e (ए), shī (शी)	-nī (नी), -hī (ही), e (ए), shī (शी)
caturthī (चतुर्थी)	Fourth	sampradāna(सम्प्रदान)	Dative case	-sa (-स), -lā (-ला), -te (-ते)	-sa (-स), -lā (-ला), -nā (ना), -te (-ते)
pancamī (पञ्चमी)	Fifth	apādāna(अपादान)	Ablative case	-ūna (-ऊन), -hūna (-हून)	-ūna (-ऊन), -hūna (-हून)
shhashhthī (षष्ठी)	Sixth	sambandh (संबंध)	Genitive case	-chā (-चा), -chī (-ची), -che (-चे)	-ce (-चे), -cyā (-च्या), -cī (-ची)
saptamī (सप्तमी)	Seventh	adhikaran (अधिकरण)	Locative case	-ta (-त), -i (-इ), -ā (-आ)	-ta (-त), -ī (-ई), -ā (-आ)
sambhodan (संबोधन)	Vocative case	–	Vocative case	–	-no (-नो)

Table 1: *Vibhakti's* in Marathi

The attribute values for noun & pronoun								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
sample words	**lex**	**cat**	**gend**	**num**	**pers**	**case**	**vib**	**tam**
मनाच्या	मन	n	n	sg		o	च्या	०ा_च्या'
मनाने	मन	n	n	sg		o	ने	०ा_ने'
पर्यटनाची	पर्यटन	n	n	sg		o	ची	०ा_ची'
दुष्परिणामापासून	दुष्परिणाम	n	m	sg		o	पासून	०ा_पासून'
पर्यटकाला	पर्यटक	n	m	sg		o	ला	०ा_ला'
सौंदर्याने	सौंदर्य	n	n	sg		o	ने	०ा_ने'
लोकांसाठी	लोक	n	m	pl		o	साठी	०ा०ं_साठी'
त्यांच्यासाठी	तो	pn	m	pl		o	साठी	्यां_च्या_साठी'
ह्यांचे	हा	pn	m	pl		o	चे	्यां_चे'

Table 2: The attribute values for Noun & Pronoun

***Vibhakti* to Number Conversion**		
***Vibhakti's* (7)**		**Numerical Form**
Singular Suffixes	**Plural Suffixes**	
(एकवचन)	**(अनेकवचन)**	
–	आ	1
स ला ते	स ला ना ते	2
नी ए शी	नी ही ए शी	3
स ला ते साठी करिता	स ला ना ते साठी करिता	4
ऊन हून मुळे पासून	ऊन हून मुळे पासून	5
चा ची चे	चे च्या ची	6
त इ आ	त ई आ	7
–	नो	8

Table 3: *Vibh[akti* to Number Conversion

5 *Verbframator*: Features and Limitations

Salient features of *Verbframator:*

- *Verbframator* reduces the manual effort of creating verb frames.
- Verb frames are automatically extracted which requires minimal validation.
- It also generates SSF files of the input verb sentences.
- Helps in analyzing the generated verb frames and constructing similar patterns for a given verb.
- *verbframe* database stores verb frame and trained data in a systematic and classified manner.

Limitations of *Verbframator*

- Although *Verbframator* generates verb frames and SSF files automatically, it still requires manual intervention to complete some tasks.
- In wordnet, not all verbs in a synset are explained using an example sentence. Only first frequently used verb appears in the example sentence. The example sentence is not available for rest of the verbs in a given synsets. Hence, verb frame for such sentences cannot be developed.
- The *Verbframator* extracts examples having input verbs only.
- Quality of verb frames depends upon annotators understanding in annotating verb frames. Hence, inter-annotator-agreement should be of high value which can lead to better quality resource.
- Quality of verb frames also depends upon the quality of trained data or gold data.
- Currently *Verbframator* can be applicable only for wordnet sentences. It cannot be used for general corpus.

6 Conclusion and Future Scope

Verb is an important POS category for any language. Verb frames are created to generate dependency tree structures in a given language. This paper introduces an online tool *Verbframator* which semi-automatically extracts and generates verb frames using example sentences from Marathi wordnet. *Verbframator* also generates SSF formatted files of the given example sentences. These resources can be used in the dependency tree banking. In future, this verb frame annotator tool is to be made fully automatic. Also, this tool is to be expanded for other Indian languages under the umbrella of IndoWordNet. Sentences from other corpus are also to be included in the *Verbframator*.

References

Bharati, Akshar, Rajeev Sangal, and Dipti M. Sharma. 2007. "*Ssf: Shakti standard format guide.*" Language Technologies Research Centre, International Institute of Information Technology, Hyderabad, India (2007): 1-25.

Begum, Rafiya, Samar Husain, Lakshmi Bai, and Dipti Misra Sharma. 2008 "*Developing Verb Frames for Hindi." In LREC*. LREC 2008.

Ghosh, Sanjukta. 2014. "Making Verb Frames for Bangla Vector Verbs" ICON 2014.

Ghosh, Sanjukta. 2014. "*Noun Classification from the Verb Frames of Conjunct Verbs.*" Forthcoming in Journal of Advanced Linguistic Studies. Bahri Publications, New Delhi.

Schulte im Walde, Sabine. 2009. "*The induction of verb frames and verb classes from corpora.*" Corpus linguistics. an international handbook. Berlin: Mouton de Gruyter (2009): 952-972.

Soni, Ankush, Sambhav Jain, and Dipti Misra Sharma. 2013. *"Exploring Verb Frames for Sentence Simplification in Hindi."* In IJCNLP, pp. 1082-1086.

Theakston, Anna L., et al. 2015. "*Productivity of noun slots in verb frames.*"Cognitive science 39.6 (2015): 1369-1395.