Exploring Verb Frames for Sentence Simplification in Hindi

Ankush Soni **Sambhav Jain** **Dipti Misra Sharma**
Language Technologies Research Centre
IIIT Hyderabad
{ankush.soni, sambhav.jain}@research.iiit.ac.in, dipti@iiit.ac.in

Abstract

Systems processing on natural language text encounters fatal problems due to long and complex sentences. Their performance degrades as the complexity of the sentence increases. This paper addresses the task of simplifying complex sentences in Hindi into multiple simple sentences, using a rule based approach. Our approach utilizes two linguistic resources viz. verb demand frames and conjuncts' list. We performed automatic as well as human evaluation of our system.

1 Introduction

Cognitive and psychological studies, performed on 'human reading' states that the effort in reading and understanding a text increases with the sentence complexity (Klein and Kurkowski, 1974) . The modern natural language processing applications are not much different, in this respect, from humans. Processing complex sentences with high accuracy has always been a challenge in computational linguistics. This calls for techniques aiming at automatic simplification of sentences (Chandrasekar et al., 1996).

The sentence complexity can be mainly classified into 'lexical complexity' and 'syntactic complexity'. In context of natural language applications, lexical complexity can be handled significantly by utilizing various resources like lexicons, dictionary, thesaurus etc. and substitute infrequent words with their frequent counterparts (De Belder et al., 2010). To address syntactic complexity, one can analyse the structure of the sentence and then apply proper operations to simplify the structure.

There are many applications of sentence simplification in NLP applications. Machine Translation systems when dealing with highly diverge language pairs face difficulty in translating long and complex sentences.

For Parsing, it has been shown by McDonald and Nivre(2007) that syntactic parsing of long sentence and Identifying long distance dependencies is still a challenging task for modern day parsers. So, it looks intuitive to break down the sentence into smaller parts and use the simplified sentences for the task of parsing and Machine translation.

In case of Automatic summarization, after simplifying sentences, it is likely that the accuracy of sentence extraction based summarization systems improves as smaller units of information are being extracted.

We present a rule based approach for sentence simplification in Hindi. Our proposed system takes a sentence and returns a set of simple sentences, smaller in length. We have taken care to produce sentences which keep the meaning close to the original sentence.

This paper is structured as follows: In Section 2, we discuss the related works for sentence simplification. Section 3 talks about complex sentence. Section 4 describes the linguistic resources we used. In section 5, we discuss our algorithm. Section 6 outlines evaluation of the system. In section 7, results are being talked about. Section 8 gives the error analysis and in Section 9, we conclude and talk about future works in this area.

2 Related Work

Chandrasekar et al.(1996) proposed Finite state grammar and Dependency based approach for sentence simplification. Automatic induction of rules for text simplification is discussed by Chandrashekhar and Srinivas (1997). A pipelined approach for text simplification has been presented by (Siddharthan, 2002). Sudoh et al. (2010) proposed divide and translate technique to address the issue of long distance reordering for machine translation. Doi and Sumita (2003) used splitting techniques for simplifying sentences and then utilizing the output for machine translation. Poorn-

International Joint Conference on Natural Language Processing, pages 1082–1086,
Nagoya, Japan, 14-18 October 2013.

ima et al. (2011) proposed a rule based Sentence Simplification for English to Tamil Machine translation system.

Though several attempts, in the past, have been carried out for English, we find few work on other languages. We find, no reported work on sentence simplification for Hindi, which is the language under focus in our work.

3 Complex Sentence

Here we are addressing sentence complexity in the context to NLP applications, and our objective is to propose resolutions which could, in general, assist and improve the performance of the NLP systems. In general, complex sentences have more than one clause (Kachru, 2006) and these clauses are combined using connectives. In the context of dependency parsing, it has been illustrated by McDonald and Nivre(2007) that the sentence length increases the complexity of a sentence, as it is difficult to process on larger sentences. On experimenting for the Hindi language, we found that as the length of the sentence increases, number of verb chunks in the sentence also increases. Based on the above observation, we consider number of verb chunks as a criterion to define complex sentences. Also, we encounter the presence of conjuncts in long sentences and concede it as the second criterion representing a complex sentence.

To consolidate, for our approach we consider a sentence to be complex based on the following criteria:

- Criterion1 : Length of the sentence is greater than 5.
- Criterion2 : Number of verb chunks in the sentence is more than 1.
- Criterion3 : Number of conjuncts in the sentence is greater than 0.

Table 1 shows classification of a sentence based on the possible combinations of 3 criteria mentioned above.

4 Linguistic Resources

A list of conjuncts and verb frames form crucial resources for splitting a complex sentence into simple sentences.

Table 1: Classification of a sentence as simple or complex

Criterion1	Criterion2	Criterion3	Category
No	No	No	Simple
No	No	Yes	Simple
No	Yes	No	Simple
No	Yes	Yes	Simple
Yes	No	No	Simple
Yes	No	Yes	Complex
Yes	Yes	No	Complex
Yes	Yes	Yes	Complex

Table 2: verb-frame

arc-label	necessity	vibh(Case)	lex-cat	src-pos
$k1$(Doer)	mandatory	0	noun	l
$k2$(Experiencer)	mandatory	0	noun	l

4.1 Connectives and Conjuncts List

Coordinating conjuncts are used to conjoin two independent clauses. Hindi coordinating conjuncts includes (ora, $athva$, yaa, $evam$, $para$, $magara$, $lekina$, $kintu$, $parantu$, $tatha$, $jabaki$, va). On the basis of the conjuncts joining two independent clauses we split the sentence for simplification.

4.2 Verb Frames

Verb frames or verb subcategorization frames, categorizes the verb on the basis of their argument demands. For Hindi, verb frames have been discussed in Begum et al. (2008) . The verb frames show mandatory $karaka$[1] relation for a verb, i.e, the arguments of a verb. Verb demand frame is represented in a tabular form shown in Table 2. A verb frame shows :

1. *karaka* : dependency arc labels
2. *Necessity* of the argument (mandatory(m) or optional(o))
3. *Vibhakti* : post-position or the case associated with the nominal
4. *Lexical category* of the arguments.
5. *Position* of the demanded nominal with respect to verb (left(l) or right(r))

The Verb demand frames are built for the base form of a verb. The demands undergo a subsequent change based on the *tense, aspect* and *modality* (TAM) of the verb used in the sentence. The knowledge about the transformations induced on the base form of a verb by TAM is stored in

[1] *karakas* are the typed dependency labels in Computational Paninian Framework(Bharati and Sangal, 1993)

form of *transformation charts* for each distinct TAM.

5 Sentence Simplification Algorithm

We present a rule based method for simplification of complex sentences. Our approach comprises two stages. The work flow of our approach is shown in Figure 1. In the first stage, we get the structural representation of the input using shallow parser.

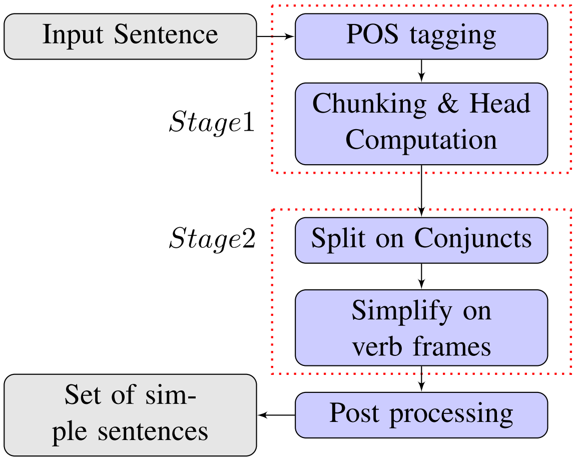

Figure 1: Flow-chart showing the work flow of sentence simplification system.

In the second stage, by applying predefined rules on the output of first stage, we identify the complexity in the sentence and simplify them on the basis of Conjuncts' list and Verb frames

5.1 Splitting on Conjuncts

In the first module, we split the sentence on the basis of Conjuncts. We identify the conjunct joining two independent clauses, break the sentence and pass it on to the second module for further simplification.

5.2 Simplification using Verb frames

After splitting the sentences on the basis of conjuncts, we simplify the generated sentences if they are complex. Once the type of sentence (complex or simple) is identified, multiple simple sentences are generated by converting non-finite verbs(VGNF) and gerunds(VGNN) to finite verb (VGF). Generally the arguments of VGNF and VGNN are shared with the main verb, therefore it is difficult for a machine to identify the implicit arguments of those verb and thus breaking the sentence and assigning arguments of those verbs explicitly helps in simplifying the sentence.

Table 3: karaka chart

arc-lbl	necessity	vibh	lex-cat	src-pos
k1	mandatory	0	noun	1
k2	mandatory	0	noun	1

For conversion of VGNF to VGF, first, the head of the chunk (VGNF) is identified using shallow parser output. Verb frame of the root form of non-finite verb is used and transformations are carried out in accordance with the TAM of the finite verb of main clause. We follow the similar procedure in case of conversion from VGNN to VGF, with a difference that pronouns are generated in the place of VGNV.
Example of VGNF:
Input:

(1) *ram khana khakar mohana ko bulata*
ram food eat-do mohana call
hai
is
'After eating food, Ram calls for Mohana'

Output:

(2) a. *ram khana khata hai*
ram food eat is
'Ram eats food'

b. *ram mohana ko bulata hai*
ram mohana calls is
'Ram calls mohana'

In Input there is a VGNF (*khakar*), and needs to be converted to VGF

Here, in Input root word of khakar is '*kha*' and TAM of VGF in main clause is '$ta - hai$'. Verb frame of '*kha*' with '*ta*' as TAM is shown in Table 3:

Here, we can see that '*kha*' has 2 requirements 'k1' and 'k2' and both should be to the left of the verb as indicated by src-pos column. So, we will look for the argument to the left of the verb and accordingly form a sentence. For the verb '*kha*' in Input, '*Ram*' will act as k1 and '*khaana*' will act as k2. Now, we are left with a finite verb '*bulata*'. For the verb - '*bulata*'in Input, *Ram* will act as 'k1' and '*Mohana ko*' will act as 'k2' with mandatory vibhakti 'ko'. As we can see here, Ram is the shared argument.

Example for VGNN

Input:

(3) $karyasthalon\ para\ anusashana$
workplaces at discipline
$banae\ rakhna\ jaruri\ hai$
maintain important is
'It is important to maintain discipline at workplaces.'

Output:

(4) a. $anusashana\ banana\ hai$
discipline maintain is
'Discipline is to be maintained'

b. $karyasthalon\ para\ yah\ behad$
workplaces at this very
$jaruri\ hai$
important is
'This is very important at workplaces.'

Here '$banae\ rakhna$' is VGNN chunk with '$banana$' as verb and '$rakhna$' as auxiliary verb.

6 Evaluation

The evaluation of sentence simplification task is a difficult problem. The evaluation should address the following two factors: Readability (Adequacy and fluency) and Simplification. To consider these factors we perform both automatic as well as human evaluation.

6.1 Data

Our testing data set consists of 100 complex sentences taken randomly from the Hindi treebank (Bhatt et al., 2009; Palmer et al., 2009).

6.2 Automatic Evaluation

We used BLEU score (Papineni et al., 2002) for automatic evaluation of our system. Higher the BLEU score, closer the target set is to the reference set. The maximum attainable value is 1 while minimum possible value is 0.

For our Automatic evaluation we adopted the same technique as Specia (2010) using BLEU metric. We performed these 3 tests:
1. Computing BLEU Score between target set and reference set.
2. Computing BLEU Score between source set and reference set.
3. Computing BLEU Score between target set and source set.

Table 4: Bleu-score for the 3 data sets

System	Gold	Bleu-score
Target	Reference	0.805
Source	Reference	0.771
Target	Source	0.750

6.3 Human Evaluation

To ensure the simplification quality subjective evaluation was done by human subjects. 20 sentences were randomly selected from the testing data-set of 100 sentences. Output of these 20 sentences, from the target set were manually evaluated by 3 subjects, who have done basic course in linguistics, for judging 'Readability' and 'Simplification' quality on the scale of $0-3$, 0 being worst to 3 being the best for readability.

For Simplification performance, scores were given according to following criteria :

- 0 = None of the expected simplifications performed.
- 1 = Some of the expected simplifications performed.
- 2 = Most of the expected simplifications performed.
- 3 = Complete Simplification.

After taking input from all the participants the results are averaged out and shown in the section 7.2.

7 Results

7.1 Automatic Evaluation

Table 4 presents the result from automatic evaluation conducted on the lines of Specia (2010).

As it is evident from the results shown, that reference set matches more to target set (**0.805**) than to source set (**0.771**). From this we can conclude that simplification performed by our system is likely to be correct.

7.2 Human Evaluation

The readability and simplification score averaged over the three subjects is **1.85** and **2.07** respectively.

8 Error Analysis

Out of the 100 sentences put to test, 61 sentences are simplified by the system. 23 cases out of the unhandled cases were already simple as per our definition in section 3 . On closer inspection we find 9 out of the remaining 16 unhandled cases are due to the presence of 'complex predicates'. Complex predicates occur in form of nominal+verb combination and thus have a generative property. Due to their generative nature it is practically challenging to create verb demand frames for them. The remaining 7 cases are found to have POS and Chunking errors. On manually evaluating the output it was found that the quality of the output is effected by the dependency relations of arguments. The verb frame cannot capture the the dependency of the required arguments thus leaving out few of the important dependencies.

9 Conclusion and Future Work

We present a rule based system for sentence simplification in Hindi. Our evaluation results show an average readability of **1.85** in the scale of 0-3, while **2.07** on the scale of 0-3 in system performance on simplification. Given the fact that this is the first attempt for Hindi we find our results satisfactory and have reason to believe that such a system will be beneficial in NLP Applications like parsing and MT. In the future our immediate effort would be on handling the complex predicates. We would like to try heuristics to capture the dependencies of the argument of verbs. We would also like to evaluate the impact of our tool on MT and parsing in the future.

Acknowledgements

We would like to thank Kunal Sachdeva, Rahul Sharma, Rajesh Chaturvedi, Rishabh Srivastava and Riyaz Ahmad Bhat for their useful comments and feedback which helped to improve this paper.

References

Rafiya Begum, Samar Husain, D Sharma, and Lakshmi Bai. 2008. Developing verb frames in hindi.

Akshar Bharati and Rajeev Sangal. 1993. Parsing free word order languages in the paninian framework. Association for Computational Linguistics.

R. Bhatt, B. Narasimhan, M. Palmer, O. Rambow, D.M. Sharma, and F. Xia. 2009. A multi-representational and multi-layered treebank for hindi/urdu.

Raman Chandrasekar and Bangalore Srinivas. 1997. Automatic induction of rules for text simplification.

Raman Chandrasekar, Christine Doran, and Bangalore Srinivas. 1996. Motivations and methods for text simplification. Association for Computational Linguistics.

Jan De Belder, Koen Deschacht, and Marie-Francine Moens. 2010. Lexical simplification.

Takao Doi and Eiichiro Sumita. 2003. Input sentence splitting and translating.

Yamuna Kachru. 2006. *Hindi.*

Gary A Klein and Frank Kurkowski. 1974. Effect of task demands on relationship between eye movements and sentence complexity.

Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models.

M. Palmer, R. Bhatt, B. Narasimhan, O. Rambow, D.M. Sharma, and F. Xia. 2009. Hindi Syntax: Annotating Dependency, Lexical Predicate-Argument Structure, and Phrase Structure.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. Association for Computational Linguistics.

C Poornima, V Dhanalakshmi, Anand M Kumar, and KP Soman. 2011. Rule based sentence simplification for english to tamil machine translation system.

Advaith Siddharthan. 2002. An architecture for a text simplification system. IEEE.

Lucia Specia. 2010. Translating from complex to simplified sentences. Springer.

Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, Tsutomu Hirao, and Masaaki Nagata. 2010. Divide and translate: improving long distance reordering in statistical machine translation. Association for Computational Linguistics.