

Exploring ways to improve Quality of Automatic Sentence Simplification for Hindi

Kshitij Mishra

IIT Hyderabad

Abstract. Sentences with high structural complexity prove to be calamitous for applications relying on natural language processing. An increasing level of intricateness or convolution in the sentence pushes the performance of NLP systems to the downgrade. These systems stand to make advantage of the process that can reduce the structural complexity of a sentence. Most of the work done on simplifying sentences of Hindi is based on rules based approaches. This work is not only limited and dependent on other resources but also prone to produce erroneous output. In this study we have first come up with improvements in one such rule based approach. The improved version of this approach is then used to produce a parallel corpus of complex and simple sentences. After manual correction, this corpus is used to perform task of sentence simplification through monolingual machine translation.

1 Introduction

Cognitive and psychological researches on human reading have revealed that greater extent of self-embedding in a sentence results in poorer learning. [1] Efforts in reading and comprehending a sentence increase with the complexity in it. In this regard, applications relying on natural language processing behave in a similar way. High structural complexity of sentences proves calamitous for the state of the art NLP applications. These systems stand to make advantage of the process that can reduce the structural complexity of the sentence. For Parsing, McDonald and Nivre(2007) [2] have shown that identifying long distance dependencies and parsing complex sentences is still a challenging task for modern day parsers. Complex syntactic structure of a sentence generates a large number of parses leading to ambiguities and errors in parsing. On the other hand, simpler sentences causes less parser ambiguities and leads to accurate parsing. Hence it seems intuitive to split a complex sentence into simpler sentences, subparses of which can be fit together forming a full parse [3]. Similarly, machine translation systems also deliver erroneous and unsatisfactory outputs for complex sentences. Simplification of sentences reduces ambiguity and improves the translation. [4] Sentence simplification can also be of great use to text summarization task as it can root out insignificant text and improves sentence extraction based summarization. Researches on spoken language understanding (SLU) systems provide evidence that sentence simplification methods are very likely to improve the

performance of these systems as they perform accurately for simple sentences whereas for complex sentences the performance degrades. [5]

(Mishra, Kshitij, et al.) [6] have showed how sentence simplification can uplift the performance of Hindi to English machine translation system. They have used a rule based approach which breaks the sentences using clause boundaries using techniques mentioned in (Sharma, Rahul, et al.) [7]. This approach works fine if there are not more than 2-3 verb chunks in the sentence. But for more complex sentences it causes loss of semantic information. [6] Moreover, their system does not change the vibhakti of the simplified sentences, which, in some cases makes the sentence lose its meaning. In this study we have come up with approaches to fix these issues and compared the results with original approach. We have also prepared a parallel corpus of complex and simple sentences which is used to perform sentence simplification using monolingual machine translation. This paper is structured as follows: In Section 2, we discuss [Mishra] approach for sentence simplification. Section 3 addresses limitations and improvements. In section 4, we discuss sentence simplification using monolingual machine translation. Section 5 outlines evaluation of the systems using and human readability . In Section 6, we conclude and talk about future work in this area.

2 Existing approaches

(Mishra, Kshitij, et al.) [6] have proposed a rule based system for sentence simplification, which first identifies the clause boundaries in the input sentence, and then splits the sentence using those clause boundaries. Once different clauses are identified, they are further processed to find shared argument for non-finite verbs. Then, the Tense-Aspect-Modality(TAM) information of the non-finite verbs is changed. The system comprises of a pipeline incorporating the following modules:

Preprocessing This module parses raw input sentences using (Jain et al., 2012) [8] dependency parser. The parser assigns a POS tag, chunk and dependency relations information in SSF format (Bharati et al., 2007) [9]

Clause boundary Identification and splitting of sentences This module takes the parsed sentence as input and identifies clause boundaries in the sentence using technique mentioned in (Sharma, Rahul, et al.) [7]. After marking the clause boundaries, the sentence is broken to simple clauses along these boundaries.

Gerunds Handler (Sharma, Rahul, et al.) identifies clause boundary only for finite and non-finite verbs. This module thus handles the gerunds in the sentence separately. (Mishra, Kshitij, et al.) have used dependency parsing information to extract the arguments of gerund and split the sentence.

Shared Argument Adder This module finds the shared argument between verbs and forms the sentence accordingly.

TAM Generator To make the split sentence more readable, TAM information of the main verb is imposed on other verbs.

3 Quality Analysis

3.1 Participants

Participants were 5 students pursuing research in the domain of Natural Language processing / Computational Linguistics. All were native Hindi speakers and proficient in Hindi. All the participants have done basic courses in linguistics.

3.2 Material

We prepared three sets of testing data, each containing 100 sentences. These sentences were taken from the Hindi treebank (Bhatt et al., 2009) [10]. The first set had sentences with 2 to 3 verb chunks the second had 4 to 5 verb chunks and the third had sentences with more than 5 verb chunks.

Table 1 shows the testing data set division.

Table 1. Testing Datasets.

Data Set	No.of verb chunks per sentence
Data Set 1	2 to 3
Data Set 2	4 to 5
Data Set 3	More than 5

3.3 Procedure

The participants were asked to rate the output for each data set on an scale of 0-3, 0 being the worst and 3 being the best. 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.

Participants were also asked to provide a feedback in case of low rating.

3.4 Results

The average ratings for data set 1 was 2.5, dataset 2 was 2.0 and dataset 3 was 1.2.

4 Major Issues

As per the feedback provided, the three main reasons of low ratings came out to be "loss of naturality/readability", "loss of semantic information" and "grammatical errors". "Loss of naturality" was tagged with more than 60% of the low rated outputs. Naturality is the natural flow of a sentence. It is the result of all the experience we have had of that language(including reading, writing, speaking and listening). Any sentence that do not fit in that frame seems anomalous to the brain and is termed as not-natural. The simplest example for explaining the naturality is gender in Hindi. Based on our training of the language, we naturally assign genders of non-living things and any deviation from our assigned gender leads to the feeling that something is wrong. In our case splitting the sentence at verb chunks containing verbs like "kaha, mana, bola, aadesh diya etc." takes away the naturality. For Example:

Ram ne mana ki wo bahut mehanati hai.
(Ram accepted that he is very hard working.)

simplified to :

Ram ne mana. (Ram accepted.)
Wo bahut mehanati hai. (He is very hard working.)

Sita ne pucha ki sooryodaya kab hoga.
(Sita asked when will the sun rise.)

simplified to:

Sita ne pucha. (Sita asked.)
Sooryodaya kab hoga. (When will the sun rise.)

In all such cases, though the output is grammatically correct, the first sentence sounds incomplete or "not-natural". Devoid of naturality also arises when the sentence is broken at each and every verb chunk. Example:

yah poochne par ki kya we dobara congress mein lautenge sangama ne kaha ki na to iski zarurat hai aur na hi peeche lautane ka sawal hi uthta hai.

(On asking whether he will return to the congress, Sangma said, it is neither required nor does the question of returning arise.)

Is simplified to:

- *Kya we dobara congress mein lautenge.* (Will he return to the congress.)

- *yah poochane par sangama ne kaha.* (When asked, Sangma said.)
- *Na to iski zarurat hai.* (Neither it is required.)
- *Na hee peeche lautana hai.* (Nor do we have to return.)
- *Iska sawal uthta hai.* (The question arises.)

It is clearly observable that the simplified sentences neither sound natural nor succeed to preserve the meaning. In few specific cases the output produced by the system were grammatically incorrect. These cases mainly arise because of the stagnancy of the case marker or *vibhakti*. (Mishra, Kshitij, et al.) have made rules to change the verb's TAM but they haven't worked on changing the case markers in the simplified sentences. Therefore sentences with disputes between verb and case marker are produced.

Example:

machharon ke katne ke baad wo beemar hue. (He got sick after the mosquitoes bites.)

Is Simplified to :

machharon ke kata. (Grammatically incorrect.)
is ke baad wo beemar hue. (After this he got sick.)

In the first simplified sentence the *vibhakti 'ke'* should have been changed to *'ne'* for the formation of a valid sentence.

Simplifying the sentences also has a tradeoff with preserving the context. In our case splitting the sentence at each and every verb chunk takes away the context which sometimes produces confusion or state of false knowledge. For example:

Ram ne jhuth bola ki Mohan mar gya hai aur Rakesh Mumbai chala gya.
 (Ram lied that Mohan was dead and Rakesh went to Mumbai.)
 is simplified to:

1. *Ram ne jhuth bola.* (Ram lied.)
2. *Mohan mar gya hai.* (Mohan was dead.)
3. *Rakesh Mumbai chala gya.* (Rakesh went to Mumbai.)

In this case all three sentences are grammatically correct but the context is lost in sentence(3). The reader, can never know whether sentence (3) is true or false.

5 Enhancements

5.1 Maintaining naturality and preserving meaning

However, naturality of a sentence is purely a subjective property, handling some profound cases can improve quality of output. As per our analysis the main reason of the loss of naturality was breaking the sentence at each and every verb chunk. For dealing with this problem we have made the following enhancements.

Not breaking at some specific verbs The evaluation of the output and the feedbacks indicate that breaking the sentence at some specific verbs destroys the flow, readability and context of the sentence. These verbs include all the "factive" verbs like "kaha, mana, bola, bataya, samjhaya, pucha etc."

Not breaking at gerunds Gerunds are verbs(verb chunks) that act as nouns in the sentence. Hence breaking a sentence at a gerund either results in loss of the meaning or makes the resulting sentence ungrammatical. Hence we decide not to break the sentence at gerunds.

Not breaking at comma separated verbs. We have decided not to break sentence at comma separated verbs.

5.2 Dealing with the *vibhakti* change

Whenever the breaking point verb is a transitive verb, the *vibhakti* will change from "ke" to "ne"

Example:

machharon ke katne ke baad wo beemar hue

(He got sick after the mosquitoes bites.)

machharon ke kata. (changes to) *machharon ne kata* (Mosquitoes bitten.)

is ke baad wo beemar hue. (After this he got sick.)

5.3 Evaluation

A similar round of evaluation was carried out with three data sets containing sentences with 2 to 3 verb chunks, 4 to 5 verb chunks and more than 5 verb chunks respectively. The average rating of the first data set almost remained unchanged whereas there was a minor improvement from 2.0 to 2.2 in the second data set. A significant improvement in the rating of the third data set was observed which jumped from 1.2 to 2.1 . For further enhancing the system a very deep knowledge of grammar and coding of more rules is required. And with more rules, more disambiguation problems arises.

6 Moving Towards Statistical Approach

Sentence simplification is the process of generating simplified version of a sentence by changing some of the lexicon material and grammatical structure of the sentence while still preserving the semantic content of the original sentence. In simple words translating from a difficult grammatical structure and lexicon to an easy one. Hence it can be seen as a task of machine translation and statistical methods of machine translation can be very useful for this task. The best thing with this approach is that it does not require a deep knowledge of grammar. It only requires data for training the model. Quality of the training data will decide the quality of the output. This fact can be used to prepare system for fulfilling different purposes. For our study we prepared the training data by manually enhancing the output of our rule based system. We used a corpus of more than 20 thousand manually corrected simplified sentences. We aim to investigate the use of phrase-based machine translation modified for the task of sentence simplification. We use the Moses software to train a PBMT model.[11]. In general, a statistical machine translation model finds a best translation e' of a text in language f to a text in language e by combining a translation model that finds the most likely translation $p(f|e)$ with a language model that outputs the most likely sentence

$$e' = \underset{e \in E}{\operatorname{argmax}} p(f|e)p(e)$$

The GIZA++ statistical alignment package is used to perform the word alignments, which are later combined into phrase alignments in the Moses pipeline [12] to build the sentence simplification model. GIZA++ utilises IBM Models 1 to 5 and an HMM word alignment model to find statistically motivated alignments between words. We invoke the GIZA++ aligner using the training simplification pairs. We run GIZA++ with standard settings and we perform no optimization. This results in a phrase table containing phrase pairs from complex sentences and simplified sentences and their conditional probabilities as assigned by Moses. Finally, we use the Moses decoder to generate simplifications for the sentences in the test set.

7 Results

Results Table- 2 shows how monolingual machine translation further improved the ratings given by participants.

8 Conclusion

In this study we carried out a detailed quality analysis of Sentence Simplification approach proposed by (Mishra, Kshitij, et al.). We have suggested some major improvements in the process to preserve the semantic context and make the simplified sentences more natural. Our enhancements have also been successful in removing some major grammatical errors from the output. We have also

Table 2. Compiled Results.

Data Set	Without Enhancements	With Enhancements	Using Machine Translation
Data Set 1	2.5	2.6	2.7
Data Set 2	2.0	2.2	2.6
Data Set 3	1.2	2.1	2.5

showed how phrase-based monolingual machine translation can be a better choice for sentence simplification. In our future work we would like to evaluate our system on NLP tasks like parsing, dialog systems, summarisation and question-answering systems.

References

1. Miller, G.A., Isard, S.: Free recall of self-embedded english sentences. *Information and Control* **7** (1964) 292–303
2. McDonald, R., Nivre, J.: Characterizing the errors of data-driven dependency parsing models. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. (2007)
3. Chandrasekar, R., Doran, C., Srinivas, B.: Motivations and methods for text simplification. In: *Proceedings of the 16th conference on Computational linguistics-Volume 2*, Association for Computational Linguistics (1996) 1041–1044
4. Chandrasekar, R.: *A Hybrid Approach to Machine Translation Using Man Machine Communication*. PhD thesis, Tata Institute of Fundamental Research (1994)
5. Tur, G., Hakkani-Tür, D., Heck, L., Parthasarathy, S.: Sentence simplification for spoken language understanding. In: *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, IEEE (2011) 5628–5631
6. Mishra, K., Soni, A., Sharma, R., Sharma, D.: Exploring the effects of sentence simplification on hindi to english machine translation system. In: *Proceedings of the Workshop on Automatic Text Simplification-Methods and Applications in the Multilingual Society (ATS-MA 2014)*. (2014) 21–29
7. Sharma, R., Paul, S., Bhat, R.A., Jain, S.: Automatic clause boundary annotation in the hindi treebank. In: *Proceedings of the 27th Pacific Asia Conference on Language, Information, and Computation (PACLIC 27)*. (2013) 499–504
8. Singla, K., Tammewar, A., Jain, N., Jain, S.: Two-stage approach for hindi dependency parsing using mltparser. *Training* **12041** (2012) 22–27
9. Bharati, A., Sangal, R., Sharma, D.M.: *Ssf: Shakti standard format guide*. Language Technologies Research Centre, International Institute of Information Technology, Hyderabad, India (2007) 1–25
10. Bhatt, R., Narasimhan, B., Palmer, M., Rambow, O., Sharma, D.M., Xia, F.: A multi-representational and multi-layered treebank for hindi/urdu. In: *Proceedings of the Third Linguistic Annotation Workshop*, Association for Computational Linguistics (2009) 186–189

11. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions, Association for Computational Linguistics (2007) 177–180
12. Och, F.J., Ney, H.: The alignment template approach to statistical machine translation. *Computational linguistics* **30** (2004) 417–449