Sentence-level Rule-Based Paraphrase Generation in Hindi

Pranav Goyal (2018114014)
International Institute of Information Technology, Hyderabad
pranav.goyal@research.iiit.ac.in

[The project can also be accessed from this GitHub link]

Abstract

In this project I propose multiple rule-based approaches to generate sentence-level paraphrases of Hindi sentences. Paraphrase generation can be essential for many application as it can be used by a computer to better interpret an input in natural language. Rule-based approach is ideal for close paraphrases without uncertainty and large-scale generation. The three methodologies discussed and implemented are: synonym/antonym replacement, word-order rearrangement and active to passive conversion of sentences. Each of the approach generated different types of paraphrases and may require certain restriction on input for the rule to be applied. The data input needs to be annotated in SSF format as other linguistic knowledge such as chunks, dependency relations were also incorporated in the rule-based approach implementation. Results were produced on the Hindi-Urdu Multi-Representational Treebank [LTRC, IIIT Hyderabad]

1. Introduction

Two sentences are paraphrases if they both represent the same proposition and thus have a common meaning. Due to which, both the sentences can entail each other and also have a common set of entailments. Generating paraphrases can be an essential component of many larger NLP applications. For example Natural Language Generation (NLG) and summarization models can use paraphrases to ensure diversity, while other models like machine translation, question answering can use paraphrases to simplify and better interpret the source text.

Paraphrasing of text can be done at multiple levels – lexical, phrase, sentence and discourse level, where every level also includes paraphrases done at lower levels. With the development in AI and machine learning, there have been many attempts towards paraphrase generators using GANs (generative adversarial networks) and machine translation models. Despite being acceptable, these systems often are unable to generate close paraphrases as the unsupervised models tend to introduce uncertainty. A rule-based approach promises much better results as predefined rules can be used to produce accurate paraphrases on large scale. [Neverilova][Carl et al.]

Much work done on paraphrase generation has been done on popular languages such as English. This area is much less explored for Indian languages like Hindi. A group of researchers from Lovely Professional University have attempted paraphrasing in Hindi using synonym and antonym replacement but most complex cases were left untouched [Sethi et al., “A Novel Approach to Paraphrase English Sentences Using Natural Language Processing”]. Although much different from languages like English, similar methodology can also be applied to generate paraphrases in Hindi using

a rule based approach. Hindi has the benefit of being a Free-word order language which allows phrase rearrangement to generate paraphrases.

This project's aim is to use a rule-based approach to generate paraphrases in Hindi. The paraphrasing will be done at sentence level as all the individual sentences will be paraphrased in isolation without the knowledge of discourse. The results will be judged on how close their meaning are to the original sentence and whether any previous information is lost or if any new sense is being added. The rules will be made general to allow generation of paraphrases on large scale.

2. **Approach**

In the project, three rule-based approaches have been explored among which only one has been previously implemented for paraphrase generation in Hindi. The rules require input to be annotated in SSF format which has been described in section 2.1. The next three section each describe the implementations of the three approaches along with the cases being handled. Some instances of the results generated by each approach have also been presented in each respective section. All three rule-based approaches have been developed in isolation of other approaches, as once fully developed their results can be easily combined using a pipeline (result of one fed into another) to generated new classes of paraphrases.

2.1 **Annotated Dataset**

The dataset used is the Hindi-Urdu Multi-Representational Treebank [LTRC, IIIT Hyderabad] which is a collection of news articles manually annotated in SSF format. [Bharati et al.] There are a total of 3497 individual sentences in the complete dataset. It is multi-representational as the discourse has been annotated for both phrase structure analysis and dependency relation based on Paninian grammar framework (karaka relations). The articles have been divided among sentences which are subdivided among chunks. The chunks are composed of individual tokens which again have been annotated for linguistic features such as root-form, gender, inflections, word tags etc. The chunks have also been marked for phrase tags such as NP, VP and dependency relations.

```
<Sentence id="4">
1 मृतकों NN <fs af='मृतक,n,m,pl,3,o,0_में,0' drel='k7:है' posn='10' vpos='vib_2' name='मृतकों' chunkId='NP' chunkType='head:NP'
cref='i13%1:t5' crefHead='मृतकों:i13' crefType='Coreference-Noun-Noun:i12'>
2 में PSP <fs af='में,psp,,,,,,' posn='20' drel='lwg__psp:मृतकों' chunkType='child:NP' name='में'>
3 एक QC <fs af='एक,num,any,any,,any,,' posn='30' drel='nmod__adj:महिला' chunkType='child:NP2' name='एक'>
4 महिला NN <fs af='महिला,n,f,sg,3,d,0,0' drel='k1:है' posn='40' name='महिला' chunkId='NP2' chunkType='head:NP2'>
5 भी RP <fs af='भी,avy,,,,,,' posn='50' drel='lwg__rp:महिला' chunkType='child:NP2' name='भी'>
6 शामिल JJ <fs af='शामिल,adj,any,any,,,,' drel='k1s:है' posn='60' name='शामिल' chunkId='JJP' chunkType='head:JJP'>
7 है VM <fs af='है,v,any,sg,3,,है,hE' stype='declarative' posn='70' voicetype='active' name='है' chunkId='VGF' chunkType='head:VGF'>
8 । SYM <fs af='।,punc,,,,,,' drel='rsym:है' posn='80' name='।' chunkId='BLK' chunkType='head:BLK'>
</Sentence>
```

Figure 1: SSF annotation of sentence 'मृतकों में एक महिला भी शामिल है।'

Along with the annotated dataset itself, a python SSF API [Bharati et al.] has been used which formats the raw annotated text files into classes such as chunks, sentences, words, nodes each with many attributes describing themselves and their relation to other classes. The implementation of the following rules-based approach each take in the individual instance of sentence class and word list (as described in SSF API) as the only parameters.

2.2 **Synonym and Antonym Replacement**

This approach paraphrases the sentence at the lexical level by replacing the individual lexical items with synonyms (words with same sense) or negated binary antonyms (words with opposite sense). The idea is that synonym replacement will not the change proposition at all if replaced word carry the same sense as the original word, while the binary antonym replacement shall negate the whole proposition (which again will be needed to re-negated to get back the original proposition). For ex "राम के कपड़े साफ हैं" will become "राम के वस्त्र साफ हैं" and "राम के कपड़े गंदे नहीं हैं" after synonym and antonym replacement respectively. To negate the whole sentence after antonym replacement, it is the governing verb (the verb chunk the originally word was an argument of) which needs to be negated. The negation of a verb phrase can simply by appending a "नहीं" in the beginning (or removing any "नहीं" if already present)

The list of synonyms and antonyms were scrapped from online website [Hindi Student]. Each entry of synonyms list formed a set where each element can be replaced by each other element in the set. The antonym list scrapped are expected to be binary antonyms which will allow their replacement to completely negate the proposition. The lists had 190 synonym sets and 750 antonym sets.

Basic implementation was to search each word in input sentence in the synonym and antonym list and replace with other elements of the set if a match was found. Some important cases included ensuring that original word shouldn't either be a proper noun or part of compound phrase. This information was extracted for the word attributes described in SSF API. Another challenge was that the synonym and antonym list contained words in root form without any inflections. Therefore replacing a word without expected inflections would result in ungrammatical sentence generation. However gender, plurality inflections are realized as suffixes therefore with the base form and inflected form of the original word the required additions and deletions can be extracted. For example to make नदियों from नदी add ियों and delete ी. If the replaced word has a similar base form, same deletions and additions were applied to it to satisfy the grammatical constraints. Therefore the synonym तटिनी would be inflected to तटिनियों before replacement. [Sangal]

Original : उन्होंने कहा कि **लड़कियों** की चिकित्सा जाँच से बलात्कार की पुष्टि हुई है ।
Paraphrase 1: उन्होंने कहा कि **बेटियों** की चिकित्सा जाँच से बलात्कार की पुष्टि हुई है ।
Paraphrase 2: उन्होंने कहा कि **नंदिनियों** की चिकित्सा जाँच से बलात्कार की पुष्टि हुई है ।

Original : दुलानी के बारे में बताया जाता है कि वह शिकार मामले में **मुख्य** गवाह है ।
Paraphrase 1: दुलानी के बारे में बताया जाता है कि वह शिकार मामले में **गौण** गवाह **नहीं** है ।

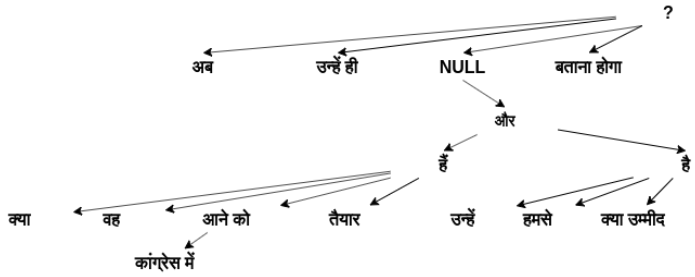
Figure 2: Examples of correct paraphrases generated by synonym and antonym replacement

2.3 **Word order rearrangement**

Word order rearrangement generates paraphrases by working at sentence-level without the knowledge of individual lexical items. Just the individual chunks and their relations are required to grammatically reorder them. Karaka relations are marked using postpositions in Hindi and, unlike English, no information is represented through the word-order itself. It is possible to generate ambiguous sentences by removing the postpositions as in "चूहे शेर खाते हैं" could mean both 'lion EAT mouse' or 'mouse EAT lion'. However examples of such unambiguous sentences were not present in the dataset which was annotated formal news articles.

Once the chunks are identified, not all the permutations are grammatical or some may not result in the original proposition. It was observed that only siblings of a dependency tree can be freely permuted as each of the sub-trees must maintain their order. This was done to ensure local coherence and sub-parts like adjectival or adverbial phrase always modified the respective head. Therefore chunks and their dependency relation were derived from the SSF API to formulate a tree, whose children could be freely permuted. As for complex sentences these permutations could generate relatively large number of paraphrases, in the final result only 5 random permutations were generated. In sentences with "कि" as the head of a K2 relation, I have replaced with a NULL character as in all cases this preserves the meaning and allows the phrase to be reordered and come before the verb phrase itself. Another observation was that the main verb can also be permuted along its modifiers, therefore it should be their sibling rather than being a parent. Therefore for all sentences, the final punctuation was swapped with the main verb to be the root of the tree. Also, before generating the tree fragof and pof tagged tokens were combined with their head to block rearrangements among them. In the final phase of paraphrase generation, for every node, its children were randomly permuted and then an inorder traversal (left children → parent → right children) of the tree was printed.

Original : अब उन्हें ही बताना होगा क्या वह कांग्रेस में आने को तैयार हैं और उन्हें हमसे क्या उम्मीद है ?
Generated tree (where all siblings can be permuted):



Random Paraphrase 1: तैयार वह क्या कांग्रेस में आने को हैं और क्या उम्मीद उन्हें हमसे है बताना होगा अब उन्हें ही ?
Random Paraphrase 2: बताना होगा उन्हें ही कांग्रेस में आने को क्या तैयार वह हैं और क्या उम्मीद उन्हें हमसे है अब ?
Random Paraphrase 2: अब बताना होगा क्या उम्मीद हमसे उन्हें है और कांग्रेस में आने को क्या वह तैयार हैं उन्हें ही ?
Random Paraphrase 2: कांग्रेस में आने को क्या तैयार वह हैं और हमसे क्या उम्मीद उन्हें है उन्हें ही अब बताना होगा ?
Random Paraphrase 2: अब बताना होगा क्या वह तैयार कांग्रेस में आने को हैं और क्या उम्मीद उन्हें हमसे है उन्हें ही ?

Figure 3: Generated tree and random paraphrases by permutation of siblings

2.4 **Active to Passive voice**

This rule-based approach again generates paraphrases by working at sentence-level by converting from active voice to passive voice. Hindi sentences can be categorize among three type of voices (वाच्य): कर्तृवाच्य, कर्मवाच्य and भाववाच्य. In the project, code for converting from कर्तृवाच्य to कर्मवाच्य has been implemented although detailed rules exist for conversions between each pair and thus can be implemented as a computer program. [Arinjay Academy] For active to passive, the K1 and K2 relations are swapped and thus the verb forms and postpositions defining the karaka relations must be modified.

For its implementation, the code for searches for sentences which contains verb phrases with distinct K1 and K2 modifiers. The verb also can not be a copula as no passive form exists for them. Once found all the postpositions were removed from the K1 and K2 chunks and the postposition "द्वारा" was inserted at the end of the K1 chunk (but before any RP words like "भी"). This was to satisfy the requirements for the arguments of passive voice of all verbs. As postpositions are combined with

pronouns as inflections, the pronouns in K1 chunks were also modified based on a predefined dictionary. Finally it was the verb form itself which was needed to be modified for passive voice. Appropriate form of "जा" was added before any auxiliaries and its inflections were derived from the verb preceding it. (there will exist a verb as only non-copula verb phrases were chosen). Also the verb was to be converted to "samanya bhootkaal" form for which it was observed that "ा" could be added to root word if ending was an consonant else "या" was to be added. Then, as the K1 and K2 relations were swapped, it was necessary to apply the new K1's gender on the verb phrase.

Original : समाजवादी मोर्चा इस ओर सरकार का ध्यान खींचने का काम करेगा ।
Paraphrase: समाजवादी मोर्चा द्वारा इस ओर सरकार का ध्यान खींचने का काम करा जाएगा ।

Original : लालू ने कहा कि 20 - 25 फीसदी निर्धनों को अभी तक फोटो पहचान - नहीं मिले हैं ।
Paraphrase 1: लालू द्वारा कहा गया कि 20 - 25 फीसदी निर्धनों को अभी तक फोटो पहचान - नहीं मिले हैं ।

Figure 4: Examples of correct paraphrases generated by active to paassive voice conversion

3. **Evaluation**

Implementation of each of the rule based approach, took individual sentence as input and returned a list of paraphrases if the respective rules were applicable. This abstraction allowed the functions to be applied to every sentence in the dataset for mass generation of paraphrases. The following table summarizes the number of sentences paraphrases by various approaches.

Approach	**Number of sentences paraphrased (out of 3497)**
Synonym replacement (without inflections)	1383
Synonym replacement (with inflections)	288
Antonym replacement (without inflections)	1812
Antonym replacement (with inflections)	649
Word-order Rearrangement	3492
Active to passive voice	1490

To find the accuracy of each of the approach, all the paraphrase could have been manually checked and classified as correct or not, however time constraint limited this. So certain limitations which can be easily observed upon going through the results have been summarized in the following sections.

3.1 **Synonym and Antonym Replacement**

The main drawback in using a synonym and antonym list scrapped from internet is that the entries are just root words and don't point to the actual sense they refer to. Because of which many ambiguous words (words with multiple possible senses) can be replaced by synonyms to a different sense of the word. Also wrong was the assumption that the list contains pure synonyms as it can be argued that it is impossible for two words to mean exactly the same in all situations. Thus often replacements resulted in completely different sense (which might even be complete nonsense if constrains on senses aren't satisfied).

The paraphrases generated by antonym replacement were poorest (in terms of the proposition applied) among all approaches as often the initial assumption (that replacement with binary antonym would completely negate the proposition) failed. Often after re-negation, instead of being a paraphrase, the result turned out to be a mere implicature of the original sentence (which also may or may not be true).

Then the trick used to deal with replacements involving inflections (by observing deletions and additions required), although mostly accurate, isn't much reliable as it isn't grammatical rule which dictates that synonyms and antonyms must share a common paradigm. [Sangal]

Original : न्यूयॉर्क में कल **सोना** ५ डॉलर बढ़कर ४२४.५० डॉलर प्रति औंस पर बंद हुआ था ।
Paraphrase: न्यूयॉर्क में कल **नींद** ५ डॉलर बढ़कर ४२४.५० डॉलर प्रति औंस पर बंद हुआ था ।

Original : शैला का चेहरा भी उसकी **खुशी** बयां कर रहा था ।
Paraphrase 1: शैला का चेहरा भी उसकी **गम** बयां **नहीं** कर रहा था ।

Figure 5: Examples of incorrect paraphrase generation with synonym/antonym replacement

3.2 **Word order rearrangement**

This approach mostly resulted in correct paraphrases because no change was required at the lexical level almost all the time, and Hindi being a free word order language allowed rearrangement of phrases in the unambiguous sentences of the news articles. The only limitation was in terms of the re-assignment and introduction of new punctuation to maintain the cohesion in the sentence.

Original : **कई कंप्यूटरों को जब जोड़ दिया गया तो स्पीड बढ़ गई ।**
Generated Paraphrase: **स्पीड जब कई कंप्यूटरों को जोड़ दिया गया तो बढ़ गई ।**
Correct Paraphrase: **स्पीड, जब कई कंप्यूटरों को जोड़ दिया गया, तो बढ़ गई ।**

Figure 6: Examples of incorrect paraphrase generation with word order rearrangement

3.3 **Active to Passive voice**

This approach too mostly generated accurate and grammatical paraphrases, however some cases were found to be not handled upon further inspection. The verb phrase must take all the properties from its subject however only gender inflections were implemented. The tricks implemented to inflect to various forms worked in most cases however again were not applicable in all the cases. For example the code had replaced "ा" with "ी" to change the gender to female but this isn't always the case. Also, pronouns in K2 were also required to be modified as the removal of postposition must involve removing the inflections from the pronouns themselves. Furthermore, it was observed that postpositions from k2 phrase were not need to be deleted in case of conjunct verb phrase.

Original :स्पीकर सोमनाथ ने भी नेताओं को समझाया।
Generated Paraphrase: स्पीकर सोमनाथ द्वारा भी नेताओं समझाया गया । (posposition was still required)

Original :अगर भारत एक कदम आगे बढ़ता है तो पाकिस्तान दो कदम आगे बढ़ाएगा ।
Generated Paraphrase: अगर भारत एक कदम आगे बढ़ता है तो पाकिस्तान द्वारा दो कदम आगे बढ़ाया जाएगा । (inflect for plurality too)

Figure 7: Examples of incorrect paraphrase generation with active to passive voice conversion

4. **Future scope**

The rules based approaches promise easy generation of sentence-level accurate paraphrases in large scale. Although rules provide certainty and precision, all the cases are needed to be explored and handled to ensure accuracy. The implementation of above approaches require input data to be in SSF format. This input although provides complete detailed about the text, provides no information to generated new ones. Because of this, the approaches which worked at sentence level were more accurate that those which required modification in lexical and phrase level. However this can be easily improved by inclusion of other components to aid modification and generation of text. A word-net or paradigm tables can be helpful to apply inflections as the root words and attributes are already known

from the analysis in SSF API. Therefore the implementation will not have to rely on morphological tricks like replacing "ा" with "ी" for changing gender.

A word-sense disambiguation model will also be helpful for generating paraphrases with synonym and antonym replacement. A major cause of error in the approach was that entries in list didn't refer to the sense being referred and thus ambiguous words were also being replaced without any further inspection. A word sense disambiguation model would allow to match and get a list of synonyms and antonyms with the specific required sense in every case.

Along with the improvements in already implemented approaches, paraphrases can also be generated by working on a higher level, specifically the discourse level. Coreference resolution can be an important component as once resolved, pronouns and other anaphora can be replaced by the predicate being referred (or others anaphora of it). Paraphrasing at discourse level can also be done by re-arrangement of sentences while maintaining the cohesion. Furthermore, various sentences can be broken down or merged together to formulate sentences with the same meaning.

5. **Conclusion**

The project has explored three effective rule-based approaches to generate sentence-level paraphrases in Hindi. Among which word-order rearrangement and active-to-passive voice conversion were most accurate as they mostly required modification at phrase and sentence level. Without a word-sense disambiguation model and paradigm tables for applying inflections, lexical-level approaches like synonym/antonym replacement worked only in general cases. As each of the approach was developed in isolation and requires only a sentence as the input, they can be easily abstracted and coupled in a pipeline to create wide variety of paraphrases. The rule-based approach to paraphrases promises accuracy and generalization for mass generation of paraphrases in Hindi.

References

[rules for active to passive] Arinjay Academy. "वाच्य परिवर्तन | Vachya Parivartan | Video | Hindi Grammar." *Arinjay Academy*, 25 Oct. 2018.

Bharati, Akshar & Sangal, Rajeev & Sharma, Dipti. (2007). Ssf: Shakti standard format guide.

Carl, Michael, et al. *Using Template-Grammars for Shake & Bake Paraphrasing*.

[antonym list] Hindi Student. "Vilom Shabd in Hindi | विलोम शब्द / विपरीतार्थक शब्द - Hindi Student." *Hindi Student*, 4 July 2017, hindistudent.com/hindi-vyakaran/vilom-shabd/vilom-shabd-in-hindi/.

[synonym list] Hindi Student. "पर्यायवाची शब्द | Paryayvachi Shabd - Hindi Student." *Hindi Student*, 27 Oct. 2018, hindistudent.com/hindi-vyakaran/paryayvachi-shabd-in-hindi/.

LTRC, IIIT Hyderabad. *A Multi-Representational and Multi-Layered Treebank for Hindi/Urdu*. *Hindi-Urdu Multi-Representational Treebanks*.

Sangal, Rajeev. "Words and Their Analyzer." Natural Language Processing - a Paninian Perspective, Prentice-Hall of India.

Neverilova, Zuzana. "Paraphrase and Textual Entailment Generation in Czech." *Computación y Sistemas*, vol. 18, no. 3, 30 Sept. 2014, 10.13053/cys-18-3-2040.

Sethi, Nandini, et al. "A Novel Approach to Paraphrase English Sentences Using Natural Language Processing." *International Science Press*.

Sethi, Nandini, et al. "A Novel Approach to Paraphrase Hindi Sentences Using Natural Language Processing." *Indian Journal of Science and Technology*, vol. 9, no. 28, 28 July 2016, 10.17485/ijst/2016/v9i28/98374.