

# Orthographic Languages Similarity Measurements

CSE-443 Natural Language Processing  
Project

Aayush Mishra (16095001)  
Kartikey Singh (16095029)

# Problem Assigned

To extract similar words between Orthographic languages along with their distance by using provided corpora with the help of Longest Common Substring (LCS) using Suffix Trees and n-gram.



# What We Did

To solve the task mentioned, we primarily used three methods:

- Suffix Tree and Longest Common Substring(LCS) matching.
- n-gram similarity measurements.
- DICE algorithm for similarity measurements.



# What is Orthography?

An orthography is a set of conventions for writing a language. It includes norms of spelling, hyphenation, capitalization, word breaks, emphasis and punctuation.

It defines the set of symbols used in writing a language, and the rules regarding how to use those symbols.



# Languages Used

In this project, we have used three pairs of orthographic languages

- Hindi and Bhojpuri
- Hindi and Magahi
- Hindi and Maithili

All these languages are part of the Indo-Aryan language family and are mostly spoken in northern and northeastern parts of India.



# Cognates

The task can also be seen as that of Cognate identification.

Cognates are words in different languages that have similar spelling and meaning.

Examples:


- action - acción (English and Spanish)
- visitor - visiteur (English and French)



# Preprocessing

The corpus provided was unfiltered and had a lot of undesired content, so to obtain a structured corpus we did some preprocessing.

The steps of preprocessing included:

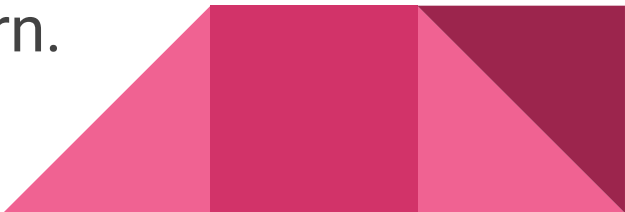
- Removing unknown files and files of very big size.
  - Removing undesired headers from some of the files.
  - Deleting any special characters and numeric characters.
  - Using Unicode to make sure only characters from Devanagari script were left.
  - Generating a list of unique words of length greater than 2 from the filtered files.
- 

After all the preprocessing, the size of the corpus for each language was:

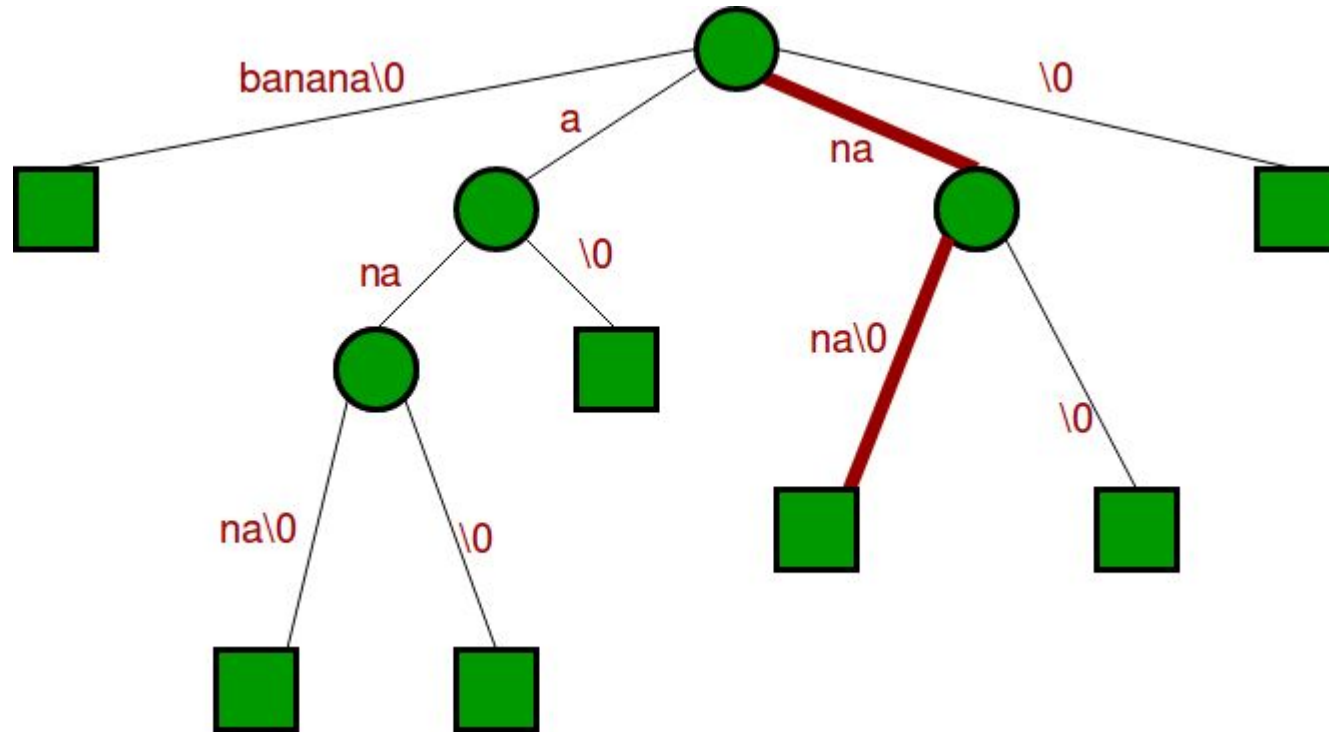
Language	Size of Corpus
Hindi	135305
Bhojpuri	36551
Magahi	33497
Maithili	25661



# Suffix Trees

- A Suffix Tree for a given text is a compressed trie for all suffixes of the given text.
  - Suffix trees are used to implement multitude of string operations faster.
  - Time taken to build a suffix tree from a text of size  $n$  is  $O(n)$  and the space used is also  $O(n)$ .
  - After the tree is built any pattern searching can be done in  $O(m)$  where  $m$  is the length of the pattern.
- 

# Suffix Tree for "banana"



# n-gram Models

- ***n-gram*** is a contiguous sequence of ***n*** items from a given sample of text or speech.
- They are used to build natural language models.
- The models are based on Markov assumption.
  - Markov models are the class of probabilistic models that assume that we can predict the probability of some future unit without looking too far in the past.



# Suffix Trees and LCS matching

To start, we first created a suffix tree from our Hindi corpus.

```
tree = SuffixTree(True , hindi_list)
```

Then picking a word from the lists of orthographic languages. We then used, the function,

```
match_list = tree.findString(word)
```

which returns to us a list of matches for that word from the Hindi suffix tree.

After that we can analyze the list to determine whether it was an Empty match, a Partial match or an exact match.

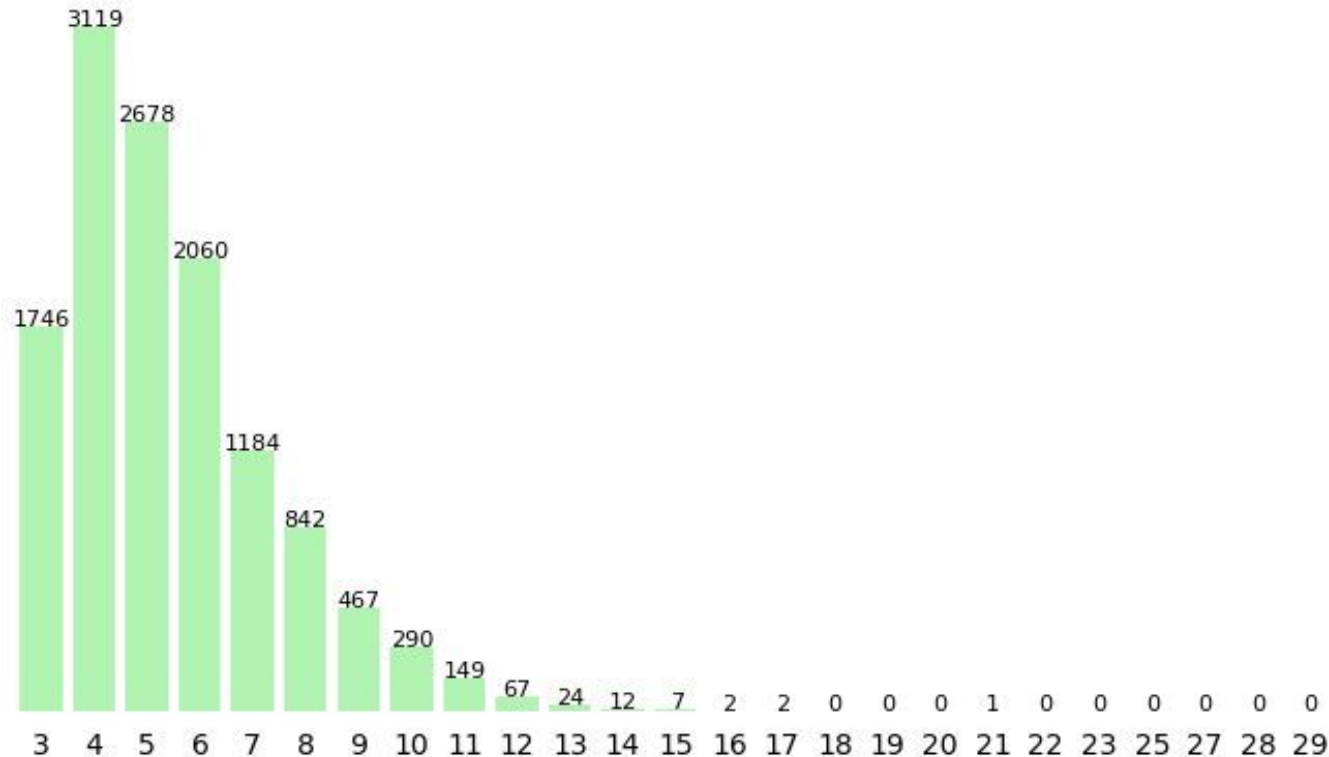


The table shows the percentage of finding a match and the percentage of finding an exact match for all the three languages.

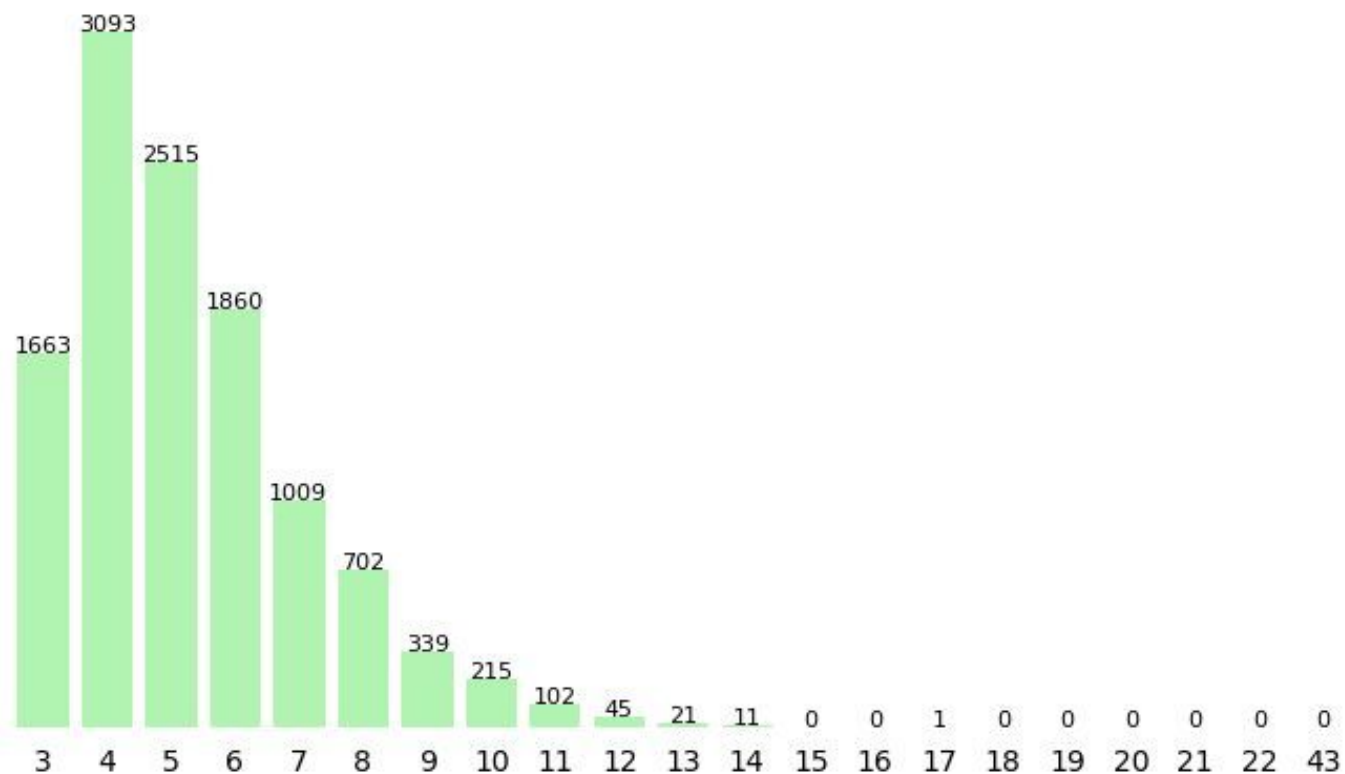
Language	Percentage of finding a match	Percentage of finding an exact match
Bhojpuri	44.269 %	34.609 %
Magahi	41.890 %	34.558 %
Maithili	39.145 %	30.228 %

The following graphs represent a relation between number of exact matches found and the length of the word for different languages

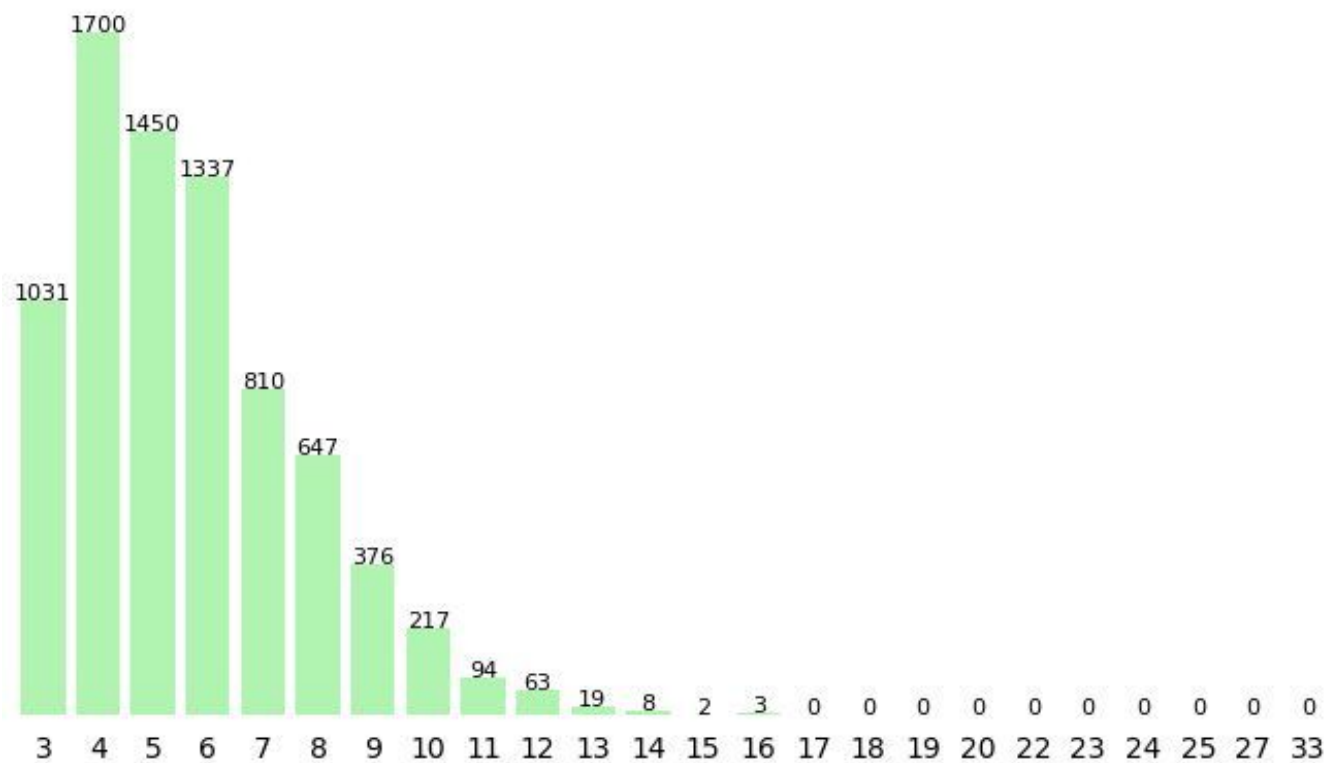
Exact Matches vs. Length of Words for Bhojpuri



Exact Matches vs. Length of Words for Magahi



Exact Matches vs. Length of Words for Maithili





# n-gram Similarity Measurement

The core idea behind n-gram similarity is to generalize the concept of the longest common subsequence to encompass n-grams, rather than just unigrams.

```
def ngram_similarity(x, y, n=1):
    k = len(x)
    l = len(y)
    L = [[0]*(l+1) for i in range(k+1)]
    for i in range(k+1):
        for j in range(l+1):
            if i == 0 or j == 0:
                L[i][j] = 0
            else:
                count = 0
                for u in range(n):
                    if i+u <= k and j+u <= l:
                        if x[i-1+u] == y[j-1+u]:
                            count += 1
                pos_ngram = (1/n)*count
                L[i][j] = max(L[i-1][j], L[i][j-1], L[i-1][j-1]+pos_ngram)
    return round(L[k][l]/max(k, l), 3)
```

The following display the best approximate match for a Hindi word in different languages along with their distance.

Hindi Word	Bhojpuri	
	Best match	Distance
निर्माणजो	नवनिर्माण	0.722
नके	अके	0.667
गदम	आदम	0.667

Hindi Word	Magahi	
	Best match	Distance
राष्ट्रपिता	राष्ट्रीयता	0.773
गीतिका	रतिका	0.667
सूर्यकुल	मूर्खाकुल	0.667

Hindi Word	Maithili	
	Best match	Distance
प्रांगण	प्रसंगक	0.643
कहानियां	ठेहुनिया	0.625
फसियो	लसिया	0.600

# DICE algorithm for similarity measurement

DICE algorithm works on the concept of character n-gram model. The basic formula that is used in this method is,

$$DICE(x, y) = 2 * |ngram(x) \cap ngram(y)| / (|ngram(x)| + |ngram(y)|)$$

For computing n-gram we used the following function:

```
def ngram(q, n=2):  
    return [q[i:i+n] for i in range(len(q)-n+1)]
```

The following display the best approximate match for a Hindi word in different languages along with their distance.

Hindi Word	Bhojpuri	
	Best match	Distance
निर्माणजो	नवनिर्माण	0.750
श्वेतकण	श्वेतकेतु	0.714
नके	सुनके	0.667

Hindi Word	Magahi	
	Best match	Distance
गारो	दारोगा	0.750
मंझोली	मंझोलका	0.727
राष्ट्रपिता	राष्ट्रीयता	0.700

Hindi Word	Mathili	
	Best match	Distance
पहाड़ियां	उड़िया	0.545
मान्यनहीं	मान्यताक	0.533
सिद्धता	सुपरसिद्ध	0.533

# Conclusion

We found similar trends for exact and partial matches for all the three languages Bhojpuri, Magahi and Maithili while using Suffix tree and LCS matching.

Most of the matches were found for word length 3-6, so we concluded that these languages are mostly made up of words of these word lengths.

But while using n-gram similarity and DICE algorithm, we found better matches for Bhojpuri and Magahi as compared to Maithili.



THANK YOU

