Bangla to Hindi Word Alignment

CDAC, Kolkata

Introduction

We already have a English to Hindi lexicon and for English - Bangla version we created a English - Bangla lexicon. So we have two lexicons in place and it was an automatic choice to device an algorithm for automatically aligning of those two lexicons, word by word, so that we get a Bangla to Hindi meaning dictionary taking the English word as a pivotal element.

The Stepwise description of the whole process

In the initial phase minimal chunk, which should be aligned structurally. The block diagram is shown in Fig.1.

Main parts of the initial phase Parsing

This is the 1st two block of initial phase. A lexicon parser was developed specifically for the AnglaBharati Lexicon format, it has two variations (i) strict parsing of the lexicon where all the formatting rules are strictly checked and any deviation is cancelled for processing. (ii) loose parsing , where all the formatting rules are not strictly checked . We opted for the second one as we did not want to miss out any information for some subtle variation of the AnglaBharati formatting structure.

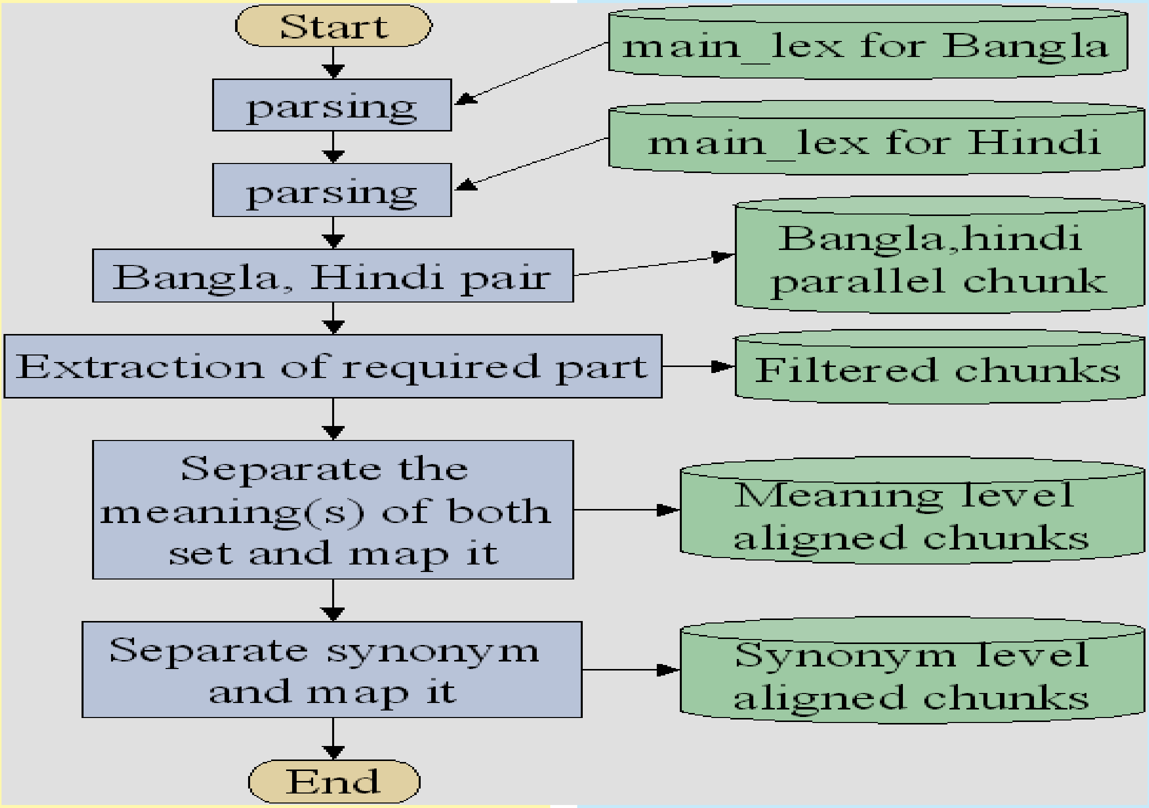
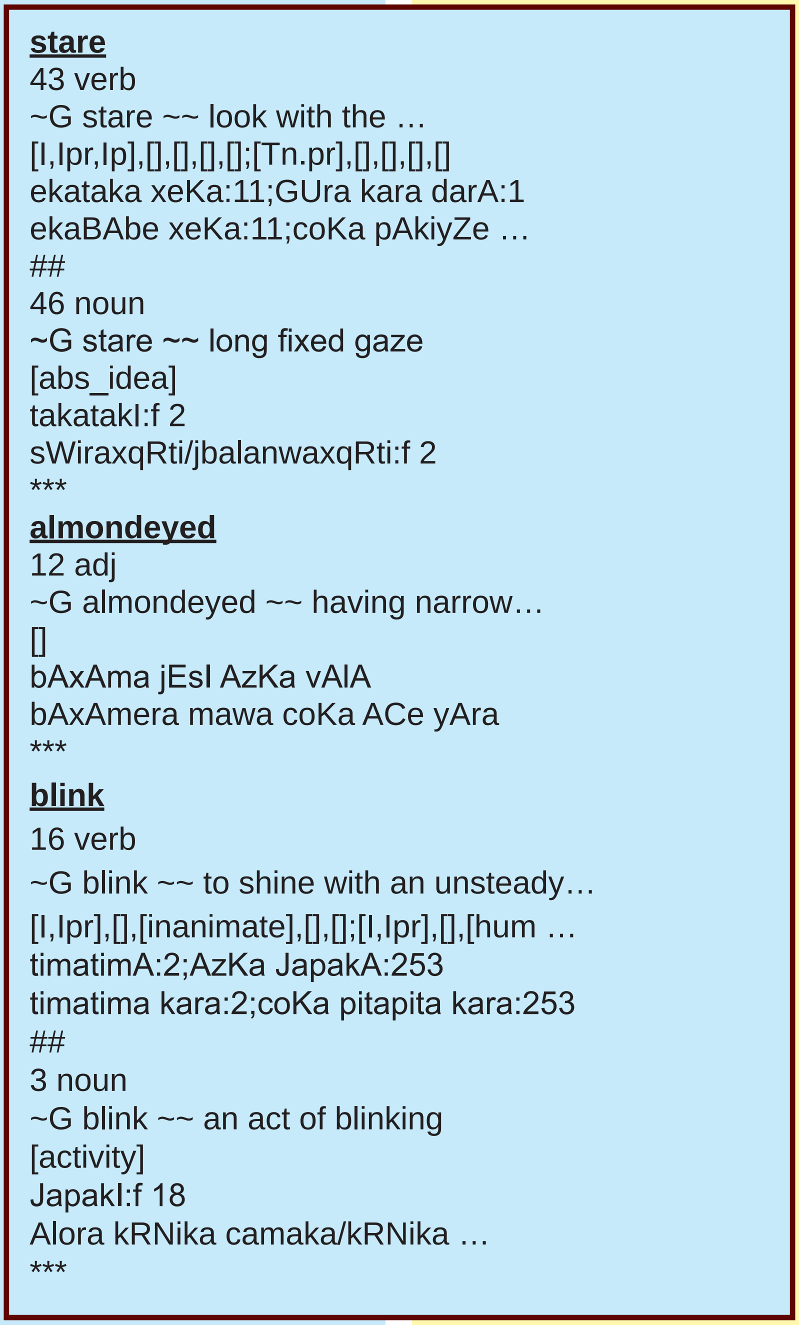

Start
main_lex for Bangla
parsing
main_lex for Hindi
parsing
Bangla, Hindi pair
Bangla,hindi parallel chunk
Extraction of required part
Filtered chunks
Separate the meaning(s) of both set and map it
Meaning level aligned chunks
Separate synonym and map it
Synonym level aligned chunks
End

Fig.1.Procedure of Filtration, Merging, Chunk Alignment as per structure.

Filtration and Structural Alignments

Firstly a union of the two lexicons, viz. English - Bangla and English - Hindi was created taking the English meaning as the pivot. After merger of the two lexicons, we became interested in the last two fields (Highlighted in Fig.4.), i.e. Romanized Bangla and Hindi, against an English word (underlined in Fig.4.). Some existing structural and spelling mistake may get reflected in the final output, which can be averted by employing strict parsing at input level and can be filtered out based on the observed error types. A snap of the output is shown in Fig.2.

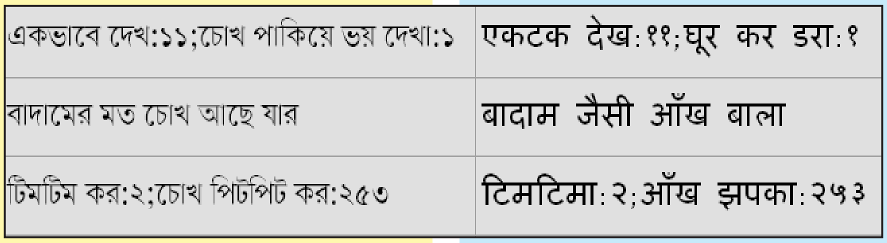


```
stare
43 verb
~G stare ~~ look with the …
[I,Ipr,Ip],[],[],[],[];[Tn.pr],[],[],[],[]
ekataka xeKa:11;GUra kara darA:1
ekaBAbe xeKa:11;coKa pAkiyZe …
##
46 noun
~G stare ~~ long fixed gaze
[abs_idea]
takatakI:f 2
sWiraxqRti/jbalanwaxqRti:f 2
***
almondeyed
12 adj
~G almondeyed ~~ having narrow…
[]
bAxAma jEsI AzKa vAlA
bAxAmera mawa coKa ACe yAra
***
blink
16 verb
~G blink ~~ to shine with an unsteady…
[I,Ipr],[],[inanimate],[],[];[I,Ipr],[],[hum …
timatimA:2;AzKa JapakA:253
timatima kara:2;coKa pitapita kara:253
##
3 noun
~G blink ~~ an act of blinking
[activity]
JapakI:f 18
Alora kRNika camaka/kRNika …
***
```

Fig.2. Merged Bangla Hindi lexicon snap with associated fields. Long lines are trimmed and shown as "…"

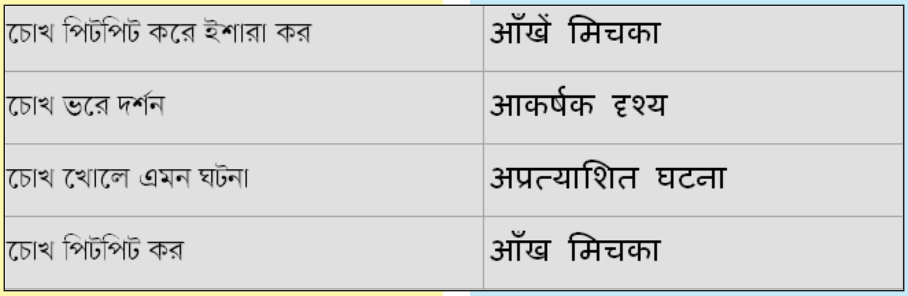
Secondly, after getting the parsed output, a memory representation of the whole structure of the lexicon is being created and it is maintained as a dictionary containing the key as English and the corresponding values as list of lists. The list of lists is a dynamically varying field as it is not fixed one for all entries of the dictionary. Also here we employed our Roman to Unicode converter to generate the Unicode representation of the Bangla and Hindi meaning for a particular English word. A snap of the output from this process is shown in Fig.3.

Now as the last step of the initial process, the entries are splitted and separated based on semicolon (';') (ref. Dictionary Structure Fig.4.) for both Bangla and Hindi. If after splitting, the number of entries (count) for both the entities (Bangla and Hindi) are same then we keep all the entries in the set of workable entries.If the count differs, then the entries are maximally matched and retained in the workable set and any additional entry(s) is separated out and treated as non-workable. So the following three files are produced (i) All matched entries for Bangla Hindi pair. (ii) Partial matched entries for Bangla Hindi pair. (iii) Unmatched entries (not used) for Bangla Hindi pair. After this, the two sets (i) and (ii) are merged together which is treated as the output from this step. Here the separated entries are again splitted and separated using front slash ('/') and the same process of matching (accepting / rejecting) and merging is repeated. Output from this step consists of 50,731 such Bangla Hindi pair.



একভাবে দেখ:১১;চোখ পাকিয়ে ভয় দেখা:১	एकटक देख:११;घूर कर डरा:१
বাদামের মত চোখ আছে যার	बादाम जैसी आँख बाला
টিমটিম কর:২;চোখ পিটপিট কর:২৫৩	टिमटिमा:२;आँख झपका:२५३

Fig.3. A snap of First level alignment of Bengali Hindi for English word "eye" as pivot



চোখ পিটপিট করে ইশারা কর	आँखें मिचका
চোখ ভরে দর্শন	आकर्षक दृश्य
চোখ খোলে এমন ঘটনা	अप्रत्याशित घटना
চোখ পিটপিট কর	आँख मिचका

Fig. 4. A snap of final alignment of Bengali Hindi for English word "eye" as pivot after initial processing

The process is depicted in the Fig.1. And the snap of the output is shown in Fig 4.

The main Parts of the final phase is described below:
The block diagram of the final phase where statistical measure is employed is described in Fig.5, 6 and 7.

The input of this phase is a series of paired lists, each consisting of Bangla and Hindi words as entries.

The Bangla list is represented as

L_B [$Bangla_1$, $Bangla_2$ ...... $Bangla_M$]

having M no. of words and the corresponding

Hindi list is

L_H [$Hindi_1$, $Hindi_2$ ...... $Hindi_N$]

having N no. of words.

Against each aligned pair, we calculate MxN distance matrix (details shown in Fig.8) as described below (Distance Measure). Then, after each calculation it must be merged with a global $M_g x N_g$ matrix with proper updates (details shown in Fig.9) described later (Merging process).

Distance Measure:_The main philosophy behind this measure is that, we suspect each Bangla word to have the correspondence with any Hindi word in that pair, but with different weight, whose formulation is motivated by the following obserZvations:

1) Each pair is most of the time left or right aligned. Left aligned means $Bangla_1$ correspondence with $Hindi_1$ and so on. Right aligned means Bangla correspondence with $Hindi_N$ and so on.
2) It has been observed that many a times Bangla and Hindi words are almost similar.

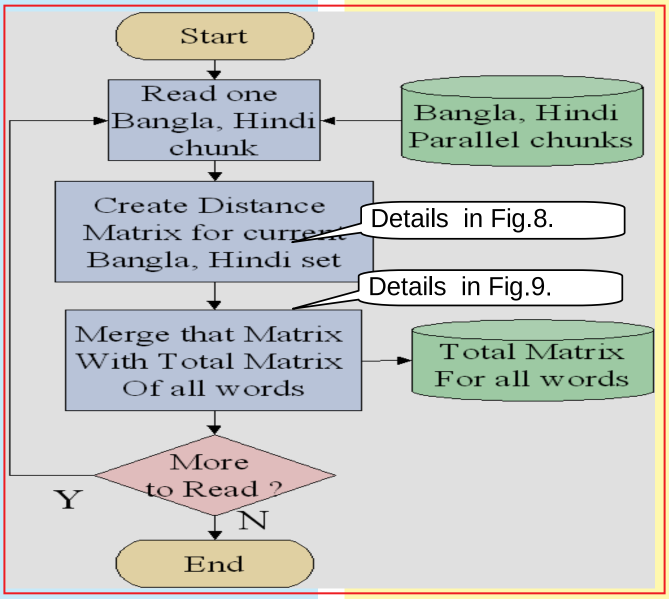

Start
Read one Bangla, Hindi chunk
Bangla, Hindi Parallel chunks
Create Distance Matrix for current Bangla, Hindi set
Details in Fig.8.
Details in Fig.9.
Merge that Matrix With Total Matrix Of all words
Total Matrix For all words
More to Read ?
Y
N
End

Fig. 5. Word level possible alignment(s) based on distance measure

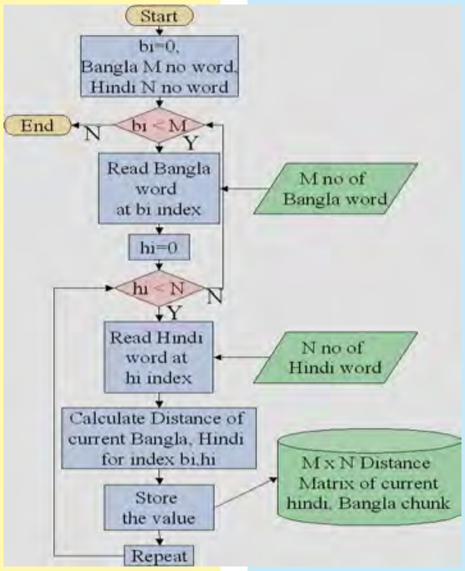


Fig. 6. Distance measure calculation against each pair of aligned Bangla Hindi words.

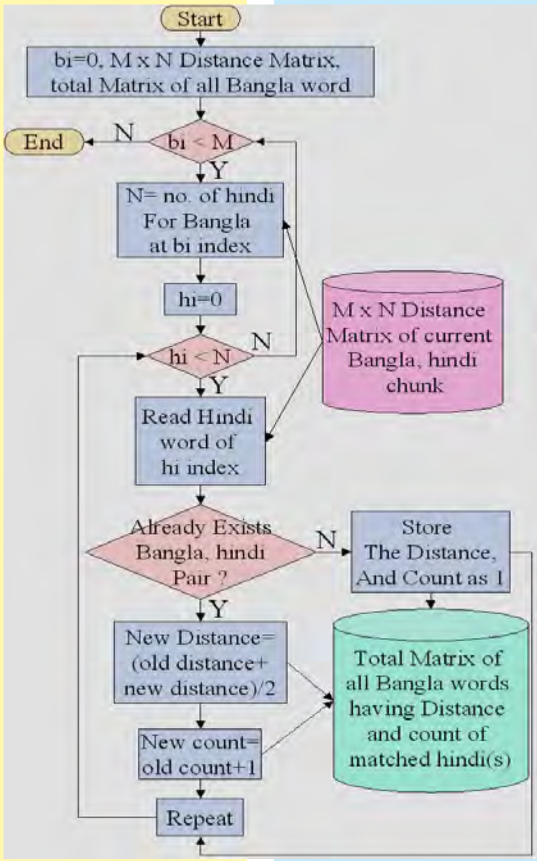
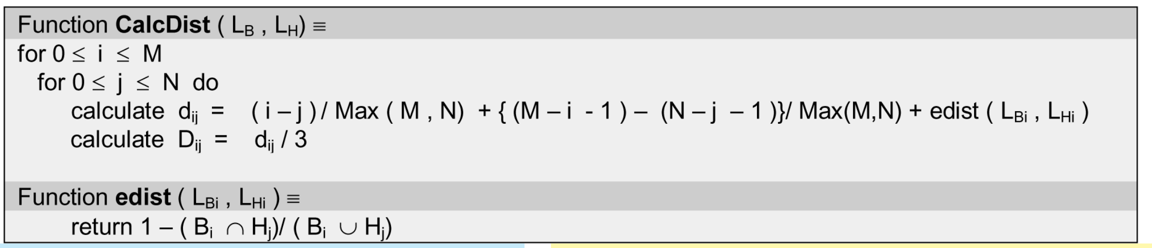


Fig.7 Merging each pair of Bangla Hindi Distance set with global Results

So our calculation for similarity measure between the words of the Bangla and Hindi entries is the average of three major components: (i) Left aligned Positional Distance, (ii) Right aligned Positional Distance, (iii) The measure of character closeness for similarity. The distance measure is shown in the following algorithm (Fig.8).



```
Function CalcDist ( $L_B$ , $L_H$) ≡
for $0 \leq i \leq M$
  for $0 \leq j \leq N$ do
      calculate $d_{ij}$ = ( i – j ) / Max ( M , N) + { (M – i - 1 ) – (N – j – 1 )}/ Max(M,N) + edist ( $L_{Bi}$ , $L_{Hi}$ )
      calculate $D_{ij}$ = $d_{ij}$ / 3

Function edist ( $L_{Bi}$ , $L_{Hi}$ ) ≡
      return 1 – ( $B_i \cap H_j$)/ ( $B_i \cup H_j$)
```

Fig. 8. Algorithm Describing the distance measure.

Merging Process:_The input of this process is a number of possible Hindi words against each Bangla word. This type of M Bangla words are there. Each Hindi word is already assigned a distance value (calculated at the previous process) against the corresponding Bangla word.

We initialize this process with empty Global or Total matrix.

Then at each calling of this process we pick all Bangla, Hindi combination for the current pair, and check whether it is in the Global matrix or not.
If it already exists, we just update the distance value by "old value + new value / 2" and add occurrence count field by 1.

Otherwise, we just place the distance as it is and set occurrence count field as 1. The sample of final output is shown in Fig.11 In the sample output each Bangla word is aligned with all probable Hindi words found anywhere in lexicon. Each Hindi word is associated with two values. 1) Average Distance from the corresponding Bangla word, 2) Frequency of occurrence with the corresponding Bangla word. In the above figure "29_3" presents the syntax like: "distance_occurrenceCount", meaning distance is 29 and occurrence count is 3.

After this measure we can judiciously set a threshold value against two components (distance and occurrence count) individually to select the accurate Hindi word(s) for a specific Bangla word.