

Language Processing Applications Stemmer, Morphological Analyzer, Others

Md. Abul Hasnat

Center for Research on Bangla Language Processing

BRAC University




Outline

- Motivation
- Stemmer for Bangla
 - Overview
 - Algorithm
 - Results
 - Application
 - Challenges
- Morphological Analyzer for Bangla
 - Components
 - Example
 - PC-KIMMO
 - JKimmo
- Conclusion



Motivation

- Most Bangla words are inflected.
 - Application / Usage (Stemming)
 - Spelling Checker
 - Information retrieval systems.
 - Reduce dictionary size.
 - Application / Usage (Morphological Analysis)
 - Spelling Checker
 - Text to Speech
 - Grammar Checker
 - Machine translation
- 

Stemmer for Bangla Overview

- Lightweight stemmer.
- Conflates terms by suffix removal.
 - Stripping off word endings from a suffix list.
 - Longest match basis.
- Advantages:
 - Computationally inexpensive
 - Domain independent.



Algorithm

- Only strips suffixes from words other than prefixes.
- Maintain a suffix list in one file.
- Most of the cases a suffix are collection of suffixes.
 - Ex: করেছিলাম = কর + েছিলাম (ে + ছি + লাম)
 - korechilam = koro + echilam (e + chi + lam)
 - Similar to Hindi stemmer.
- 72 suffixes for verb, 22 for noun and few for adjective.
- Order the suffixes according to their length.
 - suffix with biggest length will top of suffix list.



Algorithm

- Scan input word from right to left.
- Find which suffix from the list matches the given word.
- If matches found then strips the given word to stem and root.
- Exception:
 - Can't get the actual root. Example:
 - টেনেছিলাম (tenechilam) = টেনে (tene) + ছিলাম (chilam)
 - Actual root: টানা (tana)
- Solution:
 - After stripping the suffix, we try to recover the root from a given lexicon which contains this group of roots.

Results

Word	Stem	suffix
করছিলাম Korchilam	কর Koro	ছিলাম Chilam
নড়ছেন Norchen	নড় Noro	ছেন Chen
ঠেলাচ্ছিলাম Thelacchilam	ঠেলা Thela	চ্ছিলাম Cchilam
জমাচ্ছ Jomaccho	জমা Joma	চ্ছ Ccho
মারলাম Marlam	মার Maro	লাম Lam

Table : Result of our Stemming Algorithm

Application: Spelling Checker

○ Steps:

- Detect whether it is misspelled or not.
 - look up at root words dictionary and non inflected dictionary.
 - If not found then try to find out the stem.
 - If not found then this is erroneous string.



Application: Spelling Checker

- Generate suggestions if it is misspelled.
 - Deletion:
 - Inserts all letters in Bangla language at a time one after another in all possible position in the word.
 - Insertion:
 - Deletes each letter at a time and check the remaining from a lexicon if it is a valid word.
 - Substitution:
 - Delete a character at a time, replace it by all the character in Bangla and try to match a valid word from the lexicon.
 - Transposition:
 - Swap all possible pair of characters from their position.

Modification

- Provides suggestions for the mistakes in the suffix level.
- Do the suggestion generation processes at the suffix level and try to find out the stem.



Challenges

- The performance of stemmer is close to 94% for Bangla verbs.
- Takes too much time to generate suggestions.
- Spelling is over generating few suggestions.



Morphological Analyzer



Morphological Analysis

- Primary Components
 - Generative rules
 - Engine
 - User Interface



Bangla Morphology Basics

- Bangla Words are combination of
 - Noun root (নাম প্রকৃতি)
 - Verb root (ক্রিয়া প্রকৃতি)
 - Formative Affixes (প্রত্যয়)
 - Inflections (বিভক্তি)



Verb Morphology

কাল	নাম পুরুষ		মধ্যম			উত্তম
	সাধারণ (সে)	সম্মত্যক (তিনি)	সম্মত্যক (আপনি)	সাধারণ (তুমি)	বৃহৎ (তুমি)	
বর্তমান						
সাধারণ	করে	করেন	করেন	কর	করিস্	করি
ঘটমান	করছে	করছেন	করছেন	করছ	করহিস	করছি
পূর্বাঘটিত	করেছে	করেছেন	করেছেন	করেছ	করেহিস	করেছি
অনুজ্ঞা	করুক	করুন	করুন	কর	কর্	



Morphological Analysis: Example

করলাম ->

কর+ল+াম

কর+সাধারণ+অতীত+প্রথম

করেছিলাম ->

কর+য়েছ+িল+াম

কর+পুরাঘটিত+অতীত+প্রথম

করব ->

কর+ব

কর+সাধারণ+ভবিষ্যৎ+প্রথম

Recognition

কর+ল+াম -> করলাম

কর+য়েছ+িল+াম -> করেছিলাম

কর+ব -> করব

Generation

Lexical form: কর+ল+াম

Surface form: করলাম



PC-KIMMO

- Based on Kimmo Koskenniemi's two level morphology
- Components
 - Two level rules
 - Lexicon
 - Grammar
- Limitations
 - Allow only Latin script for input and output
 - No Graphical User Interface



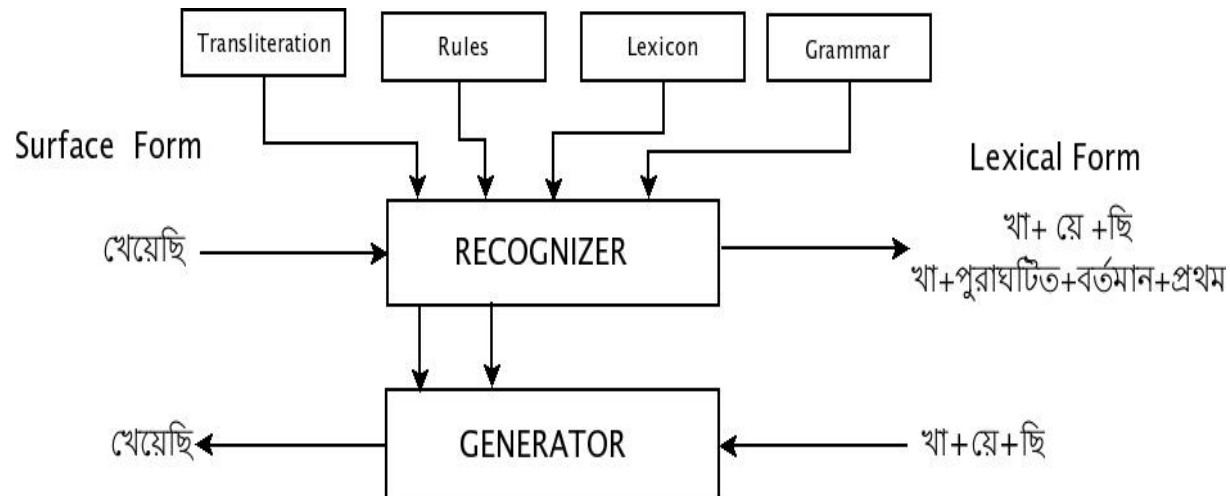
JKimmo

- A multilingual wrapper around PC-KIMMO
- Allow native language for input and output
- Localized multilingual Interface



JKimmo

- Additional component
 - Transliteration File



JKimmo components



Result & Implementation

JKimmo: Java Interface of PCKIMMO

ফাইন প্রদর্শন হাতিয়ার সহায়িকা

ব্যাকরণ লোড সংরক্ষণ সূত্র লোড সংরক্ষণ

```

<Person time> = <TRITIOGOU time>
;Rule 13
RULE
Person -> DITIOGOU
  <Person person> = <DITIOGOU person>
  <Person aspect> = <DITIOGOU aspect>
  <Person time> = <DITIOGOU time>
END
  
```

```

2: 0 3 2 3 1 1 3 1
3: 0 4 2 4 1 10 4 1
4: 0 1 2 1 5 1 1 1
5: 8 6 2 1 1 1 1 1
6: 0 1 2 1 1 7 1 1
7: 0 1 2 1 1 1 0 1
8: 0 0 0 0 0 9 0 0
9: 0 0 0 0 0 0 1 0
10: 0 4 2 4 1 1 4 1
  
```

ফলাফল সংরক্ষণ প্রদর্শনিকা লোড সংরক্ষণ

```

করেছিলাম ->
কর+য়েছ+ছিল+গেম
কর+পূর্বাঘটিত+অতিত+প্রথম
খেয়েছিলাম ->
খা+য়েছ+ছিল+গেম
খা+পূর্বাঘটিত+অতিত+প্রথম
করব ->
কর+ব
কর+সাধারণ+অবিষয়+জুড়ীয়
  
```

```

FIELDCODE lf U ;lexical item
FIELDCODE lx L ;sublexicon
FIELDCODE alt A ;alternation
FIELDCODE fea F ;features
FIELDCODE gl G ;gloss (root)

INCLUDE banglalexicon.lex

END
  
```

ইনপুট

উৎপাদন করো শব্দজ করো প্রস্থান

Conclusion

- Reusable and robust open-source framework for computational morphology
- Unicode enabled
- Multilingual Interface



Thank You

