

CSE-443

Natural Language Processing

Group 6

Sachin Kumar, 20075077

Sankalp Ranjan, 20075080

Vikram Vidhan Singh, 20074034

Problem Statement

- Converting morphological paradigms for three languages into Apertium format to build an FST-based morphological analyzer/generator, and evaluation of the Apertium-based morphological analyzer/generator
- The three Indian Languages chosen for this project were - Bhojpuri(already completed), **Magahi, and Maithili**

What is Apertium?

- Apertium is a rule-based machine translation system that was created in the project OpenTrad in 2004.
- A machine translation system is used to translate text from one language to another.
- It uses finite state transducers [FSTs] for all of its lexical transformations.

Rule-based Machine Translation (RBMT)

- Rule-based machine translation (RBMT) is an approach to machine translation that relies on explicitly defined linguistic rules and structures to translate text from one language to another.
- Broadly speaking, it involves the following process:
 - 1) Rule formation
 - 2) Constructing dictionaries from morphological paradigms
 - 3) Morphological Analysis
 - 4) Transfer
 - 5) Morphological Generation

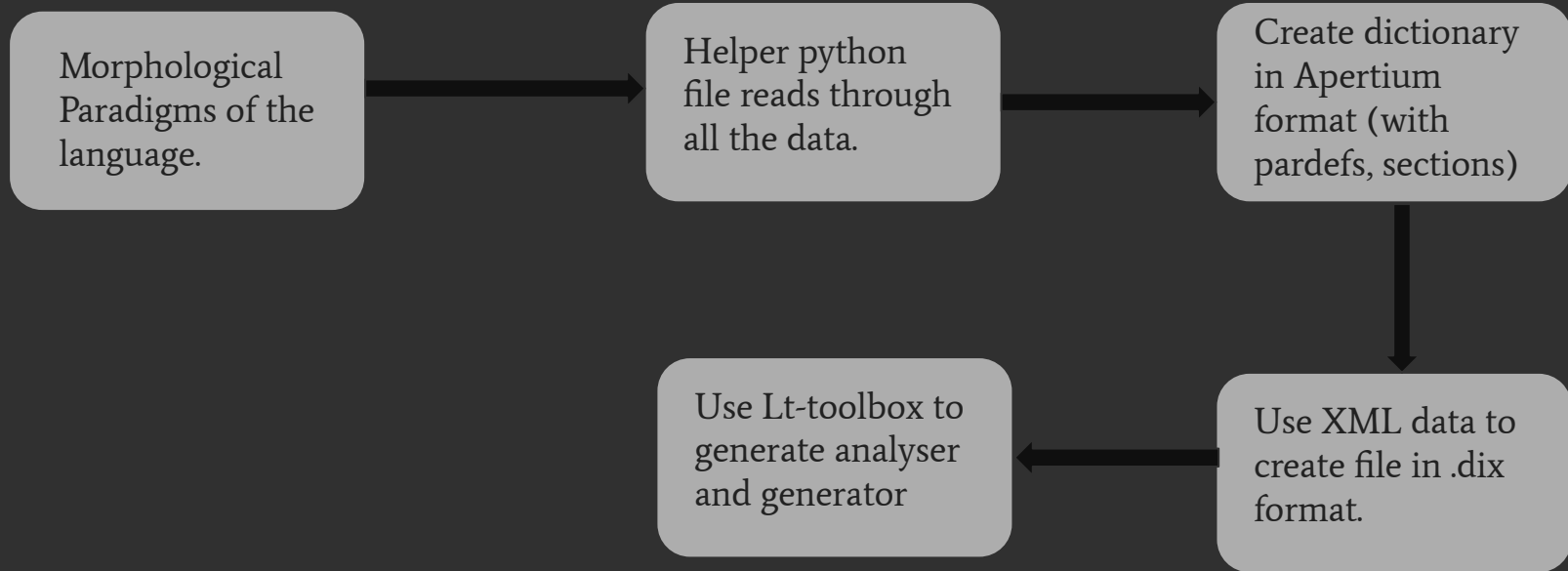
Resources Used

- Apertium framework
- Lt-toolbox
- Python 3.10
- Pycharm

Project Workflow

- We use the morphological paradigms of the language to construct dictionaries in the Apertium format.
- Once converted into the Apertium format, we can use the Lt-toolbox to generate the morphological analyzer and morphological generator from the constructed dictionary.
- We read, and construct the dictionary by reading all the morphological paradigms using a helper file (python script), and convert the XML data into .dix format to get the final dictionary in Apertium format.

Workflow (cont.)



Maithili Morphological Analyzer and Generator

```
skp_r@SankalpR: ~/nlp/Bhojpuri-Magahi-and-Maithili-Linguistic-Resources-main/maithili/...  
(base) skp_r@SankalpR:~/nlp/Bhojpuri-Magahi-and-Maithili-Linguistic-Resources-main/maithili/morphologic  
al-paradigms-and-lexicon/lexicon$ echo "^kaviye<n><m><pl><p3><o>$" | lt-proc -g maithili_generate.bin  
kaviyesaBa  
(base) skp_r@SankalpR:~/nlp/Bhojpuri-Magahi-and-Maithili-Linguistic-Resources-main/maithili/morphologic  
al-paradigms-and-lexicon/lexicon$ echo "ladZkian" | lt-proc maithili_analyze.bin  
^ladZkian/ladZki<n><f><pl><p3><o>$  
(base) skp_r@SankalpR:~/nlp/Bhojpuri-Magahi-and-Maithili-Linguistic-Resources-main/maithilili/ili/iiii  
ili/morphological-paradigms-and-lexicon/lexicon$
```


Magahi Morphological Analyzer and Generator

```
skp_r@SankalpR: ~/nlp/Bhojpuri-Magahi-and-Maithili-Linguistic-Resources-main/magahi/morph...  
(base) skp_r@SankalpR:~/nlp/Bhojpuri-Magahi-and-Maithili-Linguistic-Resources-main/magahi/morphological-paradigms-and-lexicon/lexicon$ echo "^kariya<adj><m><pl><p3><d>$" | lt-proc -g magahi_generate.bin  
kariyavA  
(base) skp_r@SankalpR:~/nlp/Bhojpuri-Magahi-and-Maithili-Linguistic-Resources-main/magahi/morphological-paradigms-and-lexicon/lexicon$ echo "kariyakko" | lt-proc magahi_analyze.bin  
^kariyakko/kariya<adj><m><sg><p3><d>$  
(base) skp_r@SankalpR:~/nlp/Bhojpuri-Magahi-and-Maithili-Linguistic-Resources-main/in/magahi/in/min/main/ii  
in/magahi/morphological-paradigms-and-lexicon/lexicon$
```

Thank You!!!

Members of Group-6,

Sankalp Ranjan

Sachin Kumar

Vikram Vidhan Singh