Transforming FLEX to CoNLL-U : Computational Challenges in Lamkang

Sadia Ashraf,

Department of Linguistics , Indiana University

Programming for Computational Linguistics

December-12-2023

This project aims at creating a CoNLL-U and investigating challenges involved in transformation from FLEX to CoNLL-U for the Lamkang language. Lamkang is a Northern Kuki-Chin Tibeto-Burman language spoken in south east of India. The corpus available in Lamkang is a valuable resource for computational analysis of the morpho-syntactic features of the low-resourced languages. The data used in this project was extracted from a partially annotated FLEX which has been provided by Professor Tyers and Professor Chelliah.

The goal of this project is to create a CoNLL-U , a standard format used in computational linguistics or natural language processing for creating computational resources such as POS taggers and treebanks. The process of creating a CoNLL-U involves getting familiar with the required components for the format which are : ID for token identification, Form for the word or symbol, Lemma as the word's base form, UPDOS for universal part-of-speech tags, XPOS for language-specific POS tags, Feats for morphological features, Head indicating the syntactic head, DepRel for the type of syntactic relation, Deps for enhanced dependency relations, and Misc for additional annotations. This project also focuses on the computational complexities when working with a low-resource language corpus and computationally non-specialized tools like FLEX.

How to run the code:

Please find the code on github with the file name “project.py”

Following steps were taken to create the CoNLL-U format:

The FLEX file was converted to xml and the library Element tree was used to parse the data. Then I created the list of all language specific tags (msa_text).

List of language -specific tags in corpus:

'v>vc', 'Inflects any category', 'vc', 'postn', 'npr', 'vc>vc', 'adn:Any', 'vt>vi', 'v:Any', 'v>dir', 'Attaches to any category', 'v>adn', 'vt>vt', 'adn>adn', 'interj', 'n>n', 'postp', 'ptc', 'vi', 'n', 'v>n', 'v>vt', 'n:Any', 'adn', 'vt>adn', 'conn', 'Particle', 'ordnum', 'dir>dir', 'dir', 'v>v', '<Not Sure>', 'vt:Any', 'cop:Any', 'num>ordnum', 'cop>cop', 'v', 'dem>dem', 'advl', 'num', 'coordconn', 'clf', 'pron', 'cop', '???>n', 'dem', 'vd', 'quant', 'n>adn', 'vt', 'subo'

A mapping function was created to map all the morpho-syntactic tags to UD POS tags.

Mapping scheme:

- v, vt, vi, vd - VERB (verb)
- n, npr, n>n, n>adn, ???>n - NOUN (noun)
- adn, adn>adn, adn:Any - ADJ (adjective)
- advl - ADV (adverb)

- ➢ interj - INTJ (interjection)
- ➢ postn, postp - ADP (adposition)
- ➢ ptc - PART (particle)
- ➢ pron - PRON (pronoun)
- ➢ num, ordnum, num>ordnum - NUM (numeral)
- ➢ cop, cop>cop, cop:Any - AUX (auxiliary)
- ➢ clf - X (other, classifier in this case)
- ➢ conn, coordconn - CONJ (conjunction)
- ➢ quant - DET (determiner, if it's quantifier-like)
- ➢ dir, dir>dir, v>dir - ADV (adverb or adposition depending on usage)
- ➢ subo - SCONJ (subordinating conjunction)
- ➢ Particle - PART (particle)
- ➢ dem, dem>dem - PRON (pronoun, demonstrative)
- ➢ v>vc, vc, vc>vc, vt>vi, v:Any, v>adn, vt>vt, v>n, v>vt, v>v, vt>adn, vt:Any, v>vc - VERB

The root was obtained and a code was written to literature through all the “paragraph” elements in the tree. These “paragraph” elements were given IDs. Also for each paragraph the text was extracted from each of it under the elements named as “words”. But only “words” followed by tags were extracted as there are “words” elements in the tree with no tags. Then for each of the “word” elements the corresponding msa_txt was printed on the same row if the number of words were not equal to the number of tags. They were printed on a new row. For each of the words, their corresponding language specific tags (msa_text) and UD POS tags were printed. While printing it was put in a CoNLL-U format.

Results:

The components extracted from the corpus were “Paragraph ID” , text , ID, word/form of word, language specific tags. Corresponding UD POS tags were used to map the language specific tags. The following elements are missing in the corpus: lemmas, head, dependency relation etc. Results showed that there are certain challenges when dealing with this corpus and it is necessary to identify and implement solutions for that. Word forms are ambiguous as words are based on one, two or even tokens so it is challenging to decide on what token ID to give to those words. Multiple language specific tags are used in the corpus which makes it challenging to ID the words and map the tags with those. Lemmas are not provided for the words. Dependency relations are unclear and no information is provided on that. Sentence segmentation can also be improved by adding sentence elements under paragraphs which are missing in this corpus. In the case of multi-word expressions it needs careful analysis on how to represent them in tagging.

Different elements can have inconsistent annotations and that can also become a challenge while creating a CoNLL-U. Despite these challenges a CoNLL-U format was successfully created based on available corpus. There is still room for improvement in the output file, for instance: implementing some kind of filter which can analyze the context and only keeps the most suitable tags for each word.

Example output:

```
# paragraph 2706
text : avah Jangma kluu humpii mii k'ool saa k'ool ava' leer ah Kokpi thung ki Koreng mii kseeng taipang
mii kseeng avah ngaa ksee chet lam dau
1       avah    _     _     PRON    _     _     _     _     dem
2       Jangma  _     _     NOUN    _     _     _     _     npr
3       kluu    _     _     VERB VERB VERB  _     _     _     _     vt:Any v>adn vt
4       humpii  _     _     NOUN    _     _     _     _     n
5       mii k'ool     _     _     NOUN VERB VERB  _     _     _     _     n v>adn vt
6       saa k'ool     _     _     NOUN VERB VERB  _     _     _     _     n v>adn vt
7       ava'    _     _     PRON    _     _     _     _     dem
8       leer    _     _     ADP     _     _     _     _     postn
9       ah      _     _     INTJ    _     _     _     _     interj
10      Kokpi   _     _     NOUN    _     _     _     _     npr
11      thung ki      _     _     ADP     _     _     _     _     postp
12      Koreng  _     _     NOUN    _     _     _     _     npr
13      mii     _     _     NOUN    _     _     _     _     n
14      kseeng  _     _     VERB VERB       _     _     _     _     v>adn vi
15      taipang _     _     NOUN    _     _     _     _     n
16      mii     _     _     NOUN    _     _     _     _     n
17      kseeng  _     _     VERB VERB       _     _     _     _     v>
```