Morphological and Syntactic Analysis

HW1 - Lexicon Acquisition Dasha Hryhoryeva

Intro

I worked with a Russian language dataset provided in the assignment. Annotation was stripped off, so I worked with a plain text file without tagging and formatting. I mostly focused on part of speech detection and partially identified nouns, verbs, adjectives, adverbs, particles, pronouns, determiners, adpositions and conjunctions. For some parts of speech, I also added class details:

- Nouns:
  - Gender – masculine, feminine, neuter
  - Declension – first, second and third
  - Example: **Tag: NOUN_MAS_2DEC**
- Verbs:
  - Conjugation – first and second
  - Reflexivity – reflexive and non-reflexive
  - Verbal form – infinitive or participle (verbal adjective), for participle there is also a gender
  - Examples: **Tag: VERB_1CONJ_INF-NREF, Tag: VERB_---_PART_MAS**
- Adjectives:
  - Gender – masculine only as it's a citation form for adjectives in Russian
  - Example: **Tag: ADG_MAS**

Regarding stem identification, I didn't use any elaborate logic, in most of the cases I removed an ending, in some cases I removed well-understood suffixes as well and called it a stem. There are also cases where I left the stem blank because of the number of various suffixes (e.g. 2-rooted nouns and adjectives). I didn't analyze prefixes since my heuristics rely on group of words with identical beginnings, and to be frank, it would significantly increase the complexity of the exercise.

The final lexicon contains around 14,000 lexemes, but can be extended if the scripts are used on a larger dataset.

Discussion

Here I'd like to shortly describe the chosen approach and heuristics.

First, I cleaned the dataset to work with alphabetical tokens only and created a frequency dictionary. From the dictionary I got the list of most frequent words. Then I removed short, rare words (with <=2 occurrences) and moved short, **frequent** (with >2 occurrences) to a separate list. For the remaining set of distinct words I identified groups with common prefixes (see create_groups_with_common_prefix function in lexicon_RL.py). Then I also moved the "lonely" words that don't have other words with the same beginning.

Static lists. I got predefined lists with POS from SynTagRus (see the Morphology section) https://universaldependencies.org/treebanks/ru_syntagrus/index.html and merged them with words from the most frequent words list I derived in the first part. I got static lists for most frequent

particles, pronouns, determiners, adpositions and conjunctions. I used these lists as a first step in applying heuristics.

Nouns. I analyzed different most common endings for nouns, e.g. -нность, -тель, -ица, -чик, -щик, -ство, -жь, -щь. I used the known rules to identify gender and declension. In some cases when I worked with plural forms, more elaborate rules were needed to identify the citation form (singular, nominative case).

Verbs

- The common ending for verbs infinitives is -ть. Verb's conjugation depends on the extended ending (e.g. -ать, -ять, -еть, -ить) plus there are some exceptions for each of the cases. I came up with a quite complicated logic to identify the conjugation for both, reflexive and non-reflexive verbs. I also used the next trick to extend the lexicon – for each non-reflexive verb I manually created a reflexive version and added it to the lexicon if it wasn't there already, and did the same when analyzed reflexive verbs.
- I also analyzed verbs in past tense with known endings -л, -ла, -ло, -ли, the tricky part was to disambiguate them from nouns with the same endings.
- Participle form of a verb I also analyzed using common endings.

Adjectives. The logic is similar to the above POS: I used all possible known endings to catch the adjectives, disambiguated them from other POS (e.g. plural ending -ий popular among nouns and adjectives) and added a masculine gender form to the lexicon. It was also tricky to add the correct ending for a masculine adjective because it depends on a voiced/voiceless consonant.

Adverbs. Nothing fancy, I again used some common endings. Typically, adverbs in Russin end with -о and are considered to be invariable, but I identified the stem based on the different types of endings ignoring the invariability assuption as this will make more sense for further analysis.

Groups of words. In cases when it was impossible to clearly identify POS based on the ending, I used the groups of words. The general approach is the following: strip off the current ending and attach all possible endings for POS that need to be checked. Then check if any of the derived inflected words are present in the group and if found, claim the word to be POS that was being checked.

Observations and summary

- For sure my heuristics don't always work correctly. I observed some number of funny examples like the below one derived from "василий", which is a male name Василий, but looks like a plural form neuter noun of type "насили**е**" (violence) and my script made up a new word.

```
Lexeme: василие          Lemma: василие          Stem: васил          Tag: NOUN_NEU_2DEC
```

- The dataset I worked with is quite small and for some rules there were simply not enough variations of lexeme to identify POS. In some cases that causes incorrect behaviour of the heuristics, in other cases the word is simply ignored and doesn't go to the final lexicon.
- It was hard to stop creating new rules! I know that probably the heuristics I used are not the most effective and I'd love to work on them longer, also, I have a bunch of ideas of what else can be analyzed.

To run the scripts

- Run **main.py** file where you can indicate INPUT_FILE_PATH and OUTPUT_FILE_PATH
- Other files include:
    - **lexicon_RL.py** prepares dataset – splitting, cleaning, creating groups with common prefixes
    - **heuristics.py** contains heuristics logic and all relevant functions
    - **lexeme.py** is a small class to store lexemes
    - **utils.py** has all the static predefined lists I'm using