Annotation Guidelines

Your role as an annotator is to check and correct model output predictions.

- Files for annotation can be found in the “cells” and “lemmas” folder for each language.
- Please annotate files one at a time in order.
- Please time yourself: exactly 30 minutes should be spent annotating cells, and 30 minutes annotating lemmas.
- In general, a higher level of discretion should be used to delink instances than when indicating additional examples you believe the model has missed.

Step 1: Delinking

- All rows labeled “True” should correspond to the same lemma or morphophonological paradigm cell.
- If you encounter a row that has been incorrectly predicted by the model as belonging to the lemma or paradigm cell in question, please add an “x” for that row to the blank annotator column (first column in the file)
- Correct instances are those belonging to the same verbal lemma or cell. Note that both overabundant and syncretic forms should be annotated as belonging to the same cell.
- Incorrect instances include those forms that are derivationally related, homophonous, or belonging to other lexical categories or (non-syncretic) paradigm cells.
- Please only mark an instance as incorrect if you are fairly confident it does not belong.

Step 2: Additional examples

Lemmas

- To help identify additional examples not predicted by the model, please run the “regex_lexeme.py” script once you have finished each lemma file.
- To ensure the syntax of the regex you would like to test is correct, you may use a website like regexr.com to check first.
- Please only test at most one regular expression per lemma.
- The programme has 5 arguments:

- --repository: the local file path for the ACL_2021 git repository
- --language: are you annotating English or Croatian? (case doesn't matter)
- --lexeme: the name of the lexeme you've just finished annotating. This is the file name without the .txt extension.
- --regex: the regex you want to generate an annotation file from. The word form should be within a capturing group.
    - For example, if you've annotated the lexeme *apply* and you only found the forms "apply" and "applying", you might want to look for instances of "applied" or even "application". The regex "(appli.+?)\\b" will pick up those occurrences. Forms that were already identified by the method as belonging to the lexeme of interest will not feature, even if they match the regex.
    - It's important that you place it between quotation marks, and that you escape the backslash.
    - The annotation file thus obtained will be in *ACL_2021/annotate/regex_output/{lexeme}.csv*
    - If no matches were found for your regex, the file will still be created. Its content is a string explaining that no matches were found.
    - If an annotation file for that lexeme already exists in the directory, the programme will add a number to the filename. This helps avoid accidentally overwriting your work.
- --n_lemma: how many occurrences do you want the programme to output? Default is 20.
- After running the script, check all grepped instances in the newly created file
- If you suspect at all that an instance does in fact contain the lemma in question, mark an “x” in the blank annotator column (first column in the file)

Paradigm Cells

- Below the set of instances predicted by the model as belonging to the cell in question (those labeled “True”) you may find a set of instances predicted to belong to other cells.
- For each of these instances, if you believe the target word form belongs to the paradigm cell in question, please change its label to “True”.