NLP Milestone

November 2024

1 Data Analysis

In the first part of the code we began with preliminary data analysis. Initially we confirmed that there are no NULL values in the dataset in the three main columns: *hyp, tgt* and *src*. Our primary goal was to create a label, so we focused the preprocessing steps on this.
The dataset is divided into three categories: **DM** (standing for definition modelling), **MT** (machine translation) and **PG** (paraphrase generation). Since each task had a different goal while model creation, we divided the dataset into three subsets accordingly.
The goal of the first task **MT** is to translate a given input into english. For this we decided to focus only on *hyp* and *tgt* to create our label, as high similarity between the hypothesis (*hyp*) and target output (*tgt*) is a good indicator for labeling, whereas the input (*src*) differs significantly due to the language difference.
For the Paraphrase Generation (**PG**) task, we tailored the preprocessing to evaluate the similarity between paraphrased sentences (*hyp*) and their sources (*src*).
For task **DM** the model should provide a short definition of the word mentioned in the description of the source. Therefore, the source only gives the context, but to judge the correctness of the definition it has to be compared to the target (*tgt*).

2 Data Preprocessing Steps

1. **Splitting text into sentences**

   Before splitting our data into sentences, we first analyzed wether this step was necessary. We examined each column individually, and found that there are for *hyp* only 243 entries with more than 1 sentence and 5 at most. In the *tgt* column we identified a total of 518 entries with more than one sentence. For *src* we got a different picture. For task DM we had more than 2000 entries with more than one sentence. Additionally

for the source input we got paragraphs with more than 10 sentences, up to a maximum of 20. For the other tasks we saw a similar picture as for the other two columns. But since the only usage for *src* is in **PG** (as said in 1) we didn't look further into that.
Since we have relatively few entries with more than one sentence, we could have skipped splitting into sentences. However some sentence combinations were conversational, with sentences acting as responses to each other. In these cases differentiation into individual sentences seems appropriate.

2. **Split text into words**

   The next step involved tokenizing the entries into individual words. We achieved this through a Stanza pipeline, which tokenized the text while maintaining information about word positions in sentences. Tokenization is essential for NLP as it converts sentences into words, which can then be further processed for understanding their syntactic and semantic roles within the context.

3. **Replacing non-words and non-whitespaces**

   We then proceeded with removing punctuation and any non-word or non-whitespace elements from the text. This removal included symbols that added no semantic value, such as music notes. Although question marks were considered important for meaning in certain cases, especially for paraphrased sentences (e.g., "Mabel's a slave?" being as a statement without a question mark), we opted to remove punctuation in general, as the overall structure and context of the sentence tend to convey hallucinations effectively.

4. **Stopword Removal**

   We also experimented with removing stopwords—common words like "and," "the," and "is"—to emphasize the core content of the text. However, we found that removing stopwords reduced accuracy, as these words often preserve essential sentence structure and meaning, particularly in paraphrased contexts. For example, removing stopwords from "it is either him or me" would leave only "either," which significantly alters the meaning. Therefore, we chose to keep stopwords in our analysis to retain the complete context.

5. **Text normalization**

   Tokenization and Lemmatization

   Using Stanza, we applied tokenization and lemmatization to transform the text into its base forms while retaining the positions of words within

the sentences. Lemmatization helps in reducing words to their root form, making it easier to analyze the text consistently.

CONLLU File and Comparison with spacy

To create this CoNLL-U file we used the function *write_doc2conll* from *stanza.utils.conll*. This gives us the text in a standardized format. For each sentence we get the information about the included words. Every row consists of the original word, its lemma, an UPOS (Universal Part-of-Speech tag) and an XPOS (Language specific part-of-speed tag. underscore if not available) as well as the position of the word in the sentence.

We also tried an alternative approach using spacy. The main difference between Stanza and spacy for generating CoNLL-U files lies in their approach and ease of use. Stanza is more straightforward for generating CoNLL-U files, aligning well with syntactic standards, while spacy offers flexibility but requires more customization to produce the same format. Also with spacy we covered head and dependency relations but Stanza was ultimately found to be a more efficient choice for this task, particularly in terms of ease of generating CoNLL-U files with minimal manual effort and more acurrancy.

3 Results of Preprocessing

3.1 DM

When looking at the ConLLU file the results look pretty good. There are some problems, especially regarding abbrevations or names. Here are some problems:

- c. as abbreviation for century is not recognized
- n in n linearly independent eigenvectors is translated to and
- New Zealand is not recognized as name but new is categorized as adjective

This happens especially often because in DM a lot of short phrases are used to provide a short definition. The sentences are often not formulated well and sometimes the words are just put after each other which makes it hard for the used pipeline to work very well with the structure of the sentence.

3.2 MT

There are cases in which the lack of apostrophes leads to incorrect word assignments in the CoNLL-U file. For example, "don't worry its only temporary" is interpreted incorrectly as "its" rather than "it is". Interestingly, when such contractions appear at the beginning of a sentence, they are sometimes handled correctly. However, without the apostrophe, words like "doesnt", "youd" or "id" are often not parsed accurately, leading to errors in both part-of-speech

tagging and lemmatization. The removal of apostrophes also impacts lemmatization, as the tool is unable to correctly derive the base form of contracted verbs. Additionally, converting all characters to lowercase results in proper nouns like "Mary" (e.g. tom told mary to tell the truth) being misinterpreted as common nouns, losing their intended context as names.

3.3 PG

The CoNLL-U file contains a few mismatches and potential errors. Some words like "cannot" could be split as "can" and "not," but "could've" appears incorrectly as "couldve" in one case. There are also some part-of-speech tagging issues, for example, "to" is labeled as a particle instead of a preposition in one sentence, and "your" might work better as a determiner rather than a pronoun in another. Parsing and dependencies show minor mismatches too, like in a sentence where "missile" directly connects to "closing" without a linking verb. Some lemma choices could be improved as well, such as considering "fifth" over "five" if the context fits, and noting that "job" appears mismatched in context.