SIL Take Home Assessment

Topic: Quality issues observed in the data along with potential methodologies for dealing with the quality issues.

Author: Robert Ferdinand

Date: February 15, 2021

Quality Issues Observed in Data:

Issue 1: The data source *BibTex* file, namely *'source.bib',* is parsed in the *PYTHON* programming language using the most optimal parsing techniques known to the author as of yet. Upon attempting to write the parsed file into a comma-separated version *(CSV)* file, one encounters *UnicodeEncodeErrors* in *PYTHON*. This is owing to the file *'source.bib*' containing text that may not be in the English language. Although this makes the data very interesting to parse, it certainly presents a challenge as regards the quality of data. One may not be able to retrieve data that contains such errors which may reduce data size and accuracy.

Issue 2: Once the data is obtained in *CSV* format, the *CSV* file contains fields from the data that have commas within themselves. This may require one to clean the *CSV* file by deleting all commas within the fields so as not to confuse the format of the comma-separated *CSV* file. This since the structure of a *CSV* file depends on the distinct fields separated by commas, necessarily.

Issue 3: The *CSV* file has fields that have unescaped quotes. That is, there is text within quotes within each field, where each field by itself is within quotes. This causes issues when one attempts to convert the *CSV* file to a *PANDAS* dataframe. The *PANDAS* dataframe is needed which is then converted to an *SQLite* database on which one can submit queries from *SQL* for example. So this is a necessary step which encounters problems owing to the unescaped quotes.

Potential Methodologies to Deal with Quality Issues:

Methodology 1: The first **Issue 1** can possibly be resolved by using or developing a parser that can recognize the characters from languages other than English and its accompanying characters and digits. That would result in the retention of more potentially useful data. Similar techniques can be developed for **Issues 2 and 3** by ignoring the extra commas and un-needed quotes, respectively, while parsing the source file. These could make the parsing and conversion to *CSV* format a lot more efficient.

Methodology 2: One may recommend creating a dictionary structure for all current languages and then label encoding (0, 1, 2, … etc.) the feature column *'lgcode'* in *'source.bib'* accordingly. This would reduce the complexity of parsing a large file significantly. Other feature columns could be label encoded as well.