Software Engineer - Backend Exercise

Duration: 2-4 hours

In this exercise, you are provided with a file (link) that contains a list of patents granted to several companies.

Each record indicates the patent number, the name of the company, and the state and country where the patent was granted.

Unfortunately, the company names contain misspellings and other noisy variations such as punctuation and legal structure indicators (e.g., INC, LLP, LLC, etc.). As a result, though the file records the patent grants for only 4 distinct companies, there are significantly more name variations contained therein.

Your exercise is to write an algorithm that can normalize the company name variations, so that we can accurately attribute the patents granted to just the 4 canonical company names: MICROSOFT TECHNOLOGY LICENSING, MICRON TECHNOLOGY, ELTA SYSTEMS, and DELTA SYSTEMS.

Requirements

- Please implement your solution in either Python or Golang.
- The script should be runnable via command line and accept a CSV file as input. It should output another CSV file that contains all the records in the original plus the normalized company name for each record.
- The normalization algorithm should be general. In other words, it should not be hardcoded for the specific company names contained in this example. If provided with another file with a different set of companies, subject to the constraints below, it should still be able to normalize the company names successfully.
    - *Hint*: Your algorithm should normalize company names for whitespace and punctuation and legal structure variations before normalizing for misspellings. You can assume that the possible legal structure suffixes (INC, LLC, etc.) are limited to the ones in the file and no misspellings are possible in the suffix.
    - *Hint*: The algorithm should not assume that the first occurrence of a company name is the canonical one. This is so that we avoid setting the canonical name to a misspelling. ○ *Hint*: You may use a fuzzy matching library to compare the similarity of two company names, but your algorithm should rely on other attributes in the file to rule out potential false positives when the names are not exactly equal. For example, fuzzy matching would say that “ELTA INC” and “DELTA INC” are highly likely the same, yet if the companies are in different locations we can consider this a false positive.
        - You can assume no misspellings are possible in the country field but *are* possible in the city field.
- Please provide your solution in a zipped source directory and include:

- Instructions on how to run the script
- Any notes on the performance, limitations, and possible improvements to your algorithm