Innovaccer
Data Analytics Assignment

M R Abhishek,
IIIT Sri city.

Technology Stack and libraries

- Python is used for creating this module.

Libraries used:

- csv : For reading the data.
- Numpy
- Levenshtein : Finding the similarity between two strings.
- random : randomize the data.

Approach

- This algorithm is an Unsupervised approach, which differentiates the names and groups into clusters.
- There are two major problems in this approach:
  - Finding whether the given two names are same or not.
  - Grouping the names.
- To achieve this task, we have to tackle the above two problems.

Problem 1: Match two names

- Two names(name_1, name_2) are splitted and stripped with ‘.’.
- The names are said to be matched if
  - At least one word in the name_1 is matched with the words in the name_2. This type of matches are counted and stored in(let’s say.. cnt_1).
  - If the word in the name_1 is not matched, then the first character is taken from the word and it is checked with the words in the name_2. This type of matches are stored in the variable.(cnt_2)
  - The matched word should also be a letter in the name_2 (Ex: M., L., ...etc).
  - A pointer is also maintained for tracking the positions. If a word is matched, the pointer is shifted to the next word and the words next to this word are only used for further comparisons.
  - This process is continued till we traverse through all the words in the name_1.

Problem 1: Match two names cont’d...

  - Also, cnt_1 + cnt_2 must also be equal to the min(len(name_1_tokens), len(name_2_tokens))
  - Return True or False based on the conditions.
- According to this algorithm, the names are matched if either of
  - match_names(name_1, name_2)
  - match_names(name_2, name_1) return True.
- The names are different only when the above two conditions returns False.
- The algorithm also matches the names that are misspelled by only one character, which is done by Levenshtein.
- This match_names function is used in clustering.

Problem 2: Group the names

- A name is taken and checked with all the names, if some of the names are matched with this name, then the names that are matched are clustered and these names are not used again.
- And this process is repeated till all the names are clustered.
- If a name is not matched with any of the other names, then the cluster consists of only one name.

Example for matching two names

- Step_1:
    - name_1 = Vladimir Frometa
    - name_2 = Vladimir F. Garo
- Step_2:
    - name_1_tokens = [‘Vladimir’, ‘Frometa’]
    - name_2_tokens = [‘Vladimir’, ‘F’, ‘Garo’]
    - cnt_1 = 0, cnt_2 = 0, pointer = 0
- Step_3:
    - The word ‘Vladimir’ is taken from name_1_tokens and checked in the name_2_tokens. As the two names are matched, we get an index ‘0’(0th position in name_2_tokens).
    - pointer = 0+1 = 1, cnt_1 = 0+1 = 1

Example cont’d...

- Step_4:
    - The word ‘Frometa’ is checked in the name_2_tokens list, we get an error as the word is not present. So, we consider the first character ‘F’.
    - The char ‘F’ is checked in the word_2_tokens[pointer:] and it is matched at index 1.
    - pointer = index + 1 = 2, cnt = 0+1 = 1
- Step_5:
    - There are no words left in the name_1_tokens. So, we stop this process here.
- Step_6: (Deciding step)
    - If cnt_1 = 0, then the names are not equal, but here cnt_1 is not equal to zero and,
    - cnt_1 + cnt_2 = 2, which is equal to min(2,3) = 2.
    - As the two conditions are satisfied, we say that the two names are same.

Example cont’d...

- Step_7:
  - The algorithm works in such a way that even if one of the match_name(name_1, name_2) or match_names(name_2, name_1) is True, then we say that the names are matched.
- Step_8:
  - In this example, the two names ‘Vladimir Frometa’ and ‘Vladimir F Garo’ are matched.

Results

- This approach is working for most of the cases.
- This algorithm is working absolutely fine on the given test set and on the example(Vladimir Frometa) given in the problem statement.
- Also achieved good results on new names.

You can find the code here.

About me

- Github: https://github.com/Abhishekmamidi123
- Blog: https://abhishekmamidi.wordpress.com
- LinkedIn: https://linkedin.com/in/abhishek-mamidi-a7a982114/