Fuzzy matching algorithm that finds the best match for company names

November 2, 2024

Algorithm selection

I chose the **Levenshtein distance (only deletion, insertion are allowed (substitution's cost is 2))** as the main part of the algorithm. My choice was influenced by many factors:

1. we need to compare just words (or sometimes just short strings consisting of 1-5 words) - company names. Therefore, the algorithms best suited for this task must somehow **compare the characters** of two strings to get their similarity coefficient.

2. algorithms that compare the characters of two strings, must quite good deal with **typing errors**.

3. **evaluation metric:** the Levenshtein distance is measured by the number of modifications, so it varies from 0 to infinity. In order to compare, which name from array A is closer to the name from array B we have to normalize the Levenshtein distance. I propose to normalize the distance by the sum of the lengths of compared names (it is the maximum number of modification that are needed to get from one name another). So this normalized distance will vary from 0 to 1.

4. In order to improve the performance of algorithm I propose to preprocess the text in our arrays A and B:lowercase all symbols, delete stopwords(such as Co, company, inc, llc, corporation ect.) leave only words (a-z, A-Z, 0-9 symbols),

Results and possible improvements

Results of algorithm without improvement may be found in the first part of file algorithm.ipynb.

The implemented algorithm applied to the preprocessed data handles typing errors, other possibilities of spelling the company name, very well. But there's one problem: It doesn't take abbreviations into account. It says that "NASA" is closer to "NatWest" than to "National Aeronautics and Space Administration." The fact that we normalized Levenshtein's distance by the sum of lengths also has a negative effect here; it prefers shorter names to longer ones. In order to overcome this problem

After Improvement

The final function which combines the Levenshtein's distance with checking of abbreviations is ǵet_simularity_coef_improved()

It is located in the same file algorithm.ipynb.

Company	Matched Name	Similarity Score
Hewlett-Packard	Hewlett Packard	1.0
	Hewlett.Packard.Company	1.0
	Hewlettt.Paccard	0.897
	helwet pacard inc	0.846
	NatWest Bank	0.4
	NatWest Markets Plc	0.387
	NatWest Markets	0.357
	NatWest Markets Securities Inc	0.316

Company	Matched Name	Similarity Score
	NatWest Group	0.308
	National Aeronautics and Space Administration	0.291
Google	GOOGLE LLC	1.0
	google inc..	1.0
	gogle	0.909
	Gooooogle	0.8
	Microsoft Corporation	0.267
	Microsoft Co.	0.267
	microsoft company	0.267
	NatWest Group	0.25
	miccro softt	0.235
	miccrosoftt	0.235
Microsoft	Microsoft Corporation	1.0
	Microsoft Co.	1.0
	microsoft company	1.0
	micr soft corporation	0.941
	miccro softt	0.9
	miccrosoftt	0.9
	NatWest Markets Securities Inc	0.4
	NatWest Markets	0.286
	GOOGLE LLC	0.267
	google inc..	0.267
NASA	National Aeronautics and Space Administration	1.0
	National Aeronautics and Space Administration (NASA)	0.889
	NatWest Markets Securities Inc	0.571
	NatWest Bank	0.533
	NatWest Markets	0.444
	NatWest Markets Plc	0.381
	NatWest Group	0.375
	miccro softt	0.333
	micr soft corporation	0.333
	helwet pacard inc	0.25
NatWest	NatWest Bank	0.778
	NatWest Group	0.737
	NatWest Markets	0.667
	NatWest Markets Plc	0.583
	National Aeronautics and Space Administration	0.545
	National Aeronautics and Space Administration (NASA)	0.5
	NatWest Markets Securities Inc	0.452
	helwet pacard inc	0.316
	Hewlett Packard	0.286
	Hewlett.Packard.Company	0.286
IBM	International Business Machines Corporation	1.0
	IBM Corporation	1.0
	miccro softt	0.4
	Microsoft Co.	0.4
	micr soft corporation	0.4
	NatWest Bank	0.4
	NatWest Markets	0.4
	NatWest Markets Plc	0.333
	NatWest Markets Securities Inc	0.333
	Microsoft Corporation	0.182

Final results

The table above presents the results of the final function. As we see it deals quite good with many problems: abbreviations, type errors, stopwords (such as Corporation, Company, LLC etc.).

In order to make it better we should expand the base of stopwords and deal with type errors in these stopwords.