Data Cleaning

Steps:

1. convert all the strings to upper cases.
2. extract the cities from the city, state strings.
3. remove the alias names from the resume data set.
4. convert the degree information to the (estimated) age when the candidates start the degree.

Matching Algorithm

The matching algorithm takes care of all the possible information (except gender) into consideration. Based upon 2 different criterion:

1. with degree information:

A. first name, last name and location(city) match

B. the estimated birth year should be larger than the actual birth year

2. without degree information

A. first name, last name and location(city) match

The reason to include the estimated birth year is to fully utilize the dataset. (e.g. It is highly unlikely for a person to start a Ph. D. degree at the age of 10) This results in a slightly low matching percentage but significantly increases the matching confidence.

Results and Analysis

with degree information

==========

total number of resume ids: = 152390

==========

total number of exact matched resume ids: = 42769 (28.0%)

==========

total number of multiple matched resume ids: = 7912 (5.2%)

==========

without degree information

==========

total number of resume ids: = 152390

==========

total number of exact matched resume ids: = 53544 (35.1%)

==========

total number of multiple matched resume ids: = 11667 (7.7%)

==========

Better Matching Algorithm

An interesting problem is how to match M person (tasks) in the resume data set to the N candidates (jobs) in the political data set. This is a very typical bipartite matching problem. A brief idea is described below:

Steps:

1. Group the (First Name, Last Name) as keys in both the resume dataset and political dataset.

2. Building the weighted bipartite using the information we have. Generate the weights (a data science problem) according to the information (i.e. the region match information, birth year and the degree start year etc.) given in the table. Higher weights mean more likely to achieve an exact match. The following table reveals the relationship between each candidates in the political dataset and the resume dataset.

	B1	B2	B3	B4
A1	10	3	0	10

A2	3	4	10	2
A3	10	3	1	1

In this example, A1 -> B4, A2-> B3, A3->A1 is an optimal matching. (Note A1 -> B1 is also a good match for A1 but this results in A3 having no better candidates to be matched.)

3. Generate the maximum matching via (modified) *Hungarian* algorithm (https://en.wikipedia.org/wiki/Hungarian_algorithm). The method is a O(N^3) complexity. The Hungarian algorithm can achieve a maximal matching based on the information provided in the table. It is also recommended to screen a base threshold to achieve high matching confidence.