Using String Similarity to Compare Job Roles

Dametreus Vincent

Fall 2020

GSA Government Contracting Roles

https://github.com/dametreusv/string_similarity_job_role_comparison

Purpose, Background & Limitations

In order to compare GSA (General Services Administration) rates with competitors, A govt. contractor has to look at rates based on Special Item Numbers (**SIN 132-51**), currently **SIN 51451S** which stands for Information Technology (IT) Professional Services. This is where IT companies fall into, specifically the following FPDS Codes:

- **D302 -** IT Systems Development Services
- **D306 -** IT Systems Analysis Services
- **D308 -** Programming Services
- **D311 -** IT Data Conversion Services

Purpose, Background & Limitations

The purpose of this project is to use job responsibilities to compare rates among companies in order to adjust for RFPs and hire candidates. Job responsibilities are more reliable than the name of the role because of the work the individual will be does rather than the title that they hold. We use string similarity to see how closely related one job is to another.

In order to do this we input the data, as is, so that the original format isn't lost from pdf and word documents to a csv format; while also showing the methods of how data was encoded. Next, we clean and standardize the data, then get it preprocessed in order to perform string similarity; a function of NLP (Natural Language Processing)

Limitations in encoding data manually, specifically unstructured and open sourced data, are:

- Tthe lack of consistency between each source of data.
- Data can be mispelled, have incorrect grammar, or have incorrect punctuation.
- Data is not consistent in all areas (role may have two different names in a single source or be misspelled).
- Data can be missing in certain areas or out of order which may cause great error (although a lot of effort has been made to make sure the correct data is inputted in the correct area).
- Data may repeat in areas it shouldn't, causing incorrect topic modeling.
- And much more.

It is important to carefully look at the consistency of data as we input them into csv format to avoid incorrect analysis.

Data & Source

- Data comes from the GSA website, which provides a detailed description of a contractor's roles, responsibilities, requirements, etc.
- The data is in pdf format and no company has a similar format.
- Data had to be input manually and converted to csv/dataframes in order to keep the integrity of the data consistent from contractor to contractor.

Gunnison Consulting Group Inc.

- Small business, Woman Owned business, Women Owned (WOSB)
- GSA eLibrary Page

In [2]:
```
# Copy and paste data to prep for conversion to CSV using another program
Computer Specialist I, 66.53, 68.19, 69.89, 71.64, 73.43
Computer Specialist II, 81.02, 83.04, 85.12, 87.24, 89.42
Expert Advisor, 259.51, 265.99, 272.64, 279.47, 286.45
Facilitator, 191.01, 195.79, 200.69, 205.70, 210.84
Programmer/Analyst I, 61.65, 63.19, 64.78, 66.40, 68.06
Programmer/Analyst II, 109.31, 112.04, 114.84, 117.71, 120.65
Project Administrator, 76.19, 78.10, 80.05, 82.06, 84.11
Project Analyst, 119.46, 122.44, 125.50, 128.63, 131.85
Project Director, 167.66, 171.85, 176.14, 180.54, 185.06
Project Manager, 170.02, 174.27, 178.62, 183.08, 187.66
Quality Assurance Specialist, 100.30, 102.81, 105.38, 108.01, 110.71
Sr. Data Engineer, 171.32, 175.61, 180.00, 184.50, 189.12
Sr. Programmer, 136.82, 140.23, 143.74, 147.34, 151.02
Sr. QA Specialist, 130.01, 133.26, 136.59, 140.01, 143.51
Sr. Security Engineer, 183.62, 188.21, 192.92, 197.74, 202.69
Sr. Subject Matter Expert, 247.17, 253.35, 259.69, 266.18, 272.83
Sr. Systems Analyst, 142.30, 145.85, 149.50, 153.24, 157.07
Sr. Systems Architect, 151.51, 155.29, 159.17, 163.15, 167.23
Subject Matter Specialist, 146.44, 150.10, 153.84, 157.69, 161.63
Systems Analyst I, 72.19, 73.99, 75.85, 77.74, 79.69
Systems Analyst II, 134.11, 137.46, 140.90, 144.42, 148.03
Systems/Data Architect, 107.81, 110.51, 113.27, 116.10, 119.00
Technical Manager, 139.50, 142.98, 146.56, 150.23, 153.98
Technical Writer, 50.21, 51.47, 52.76, 54.08, 55.43
Test Engineer, 80.24, 82.25, 84.30, 86.41, 88.56
Programmer Analyst III, 141.37, 144.91, 148.53, 152.25, 156.06
Business Analyst, 114.98, 117.85, 120.80, 123.82, 126.91
IT Specialist 86.04, 88.18, 90.39, 92.64, 94.96
Test Engineer - Level III, 131.62, 134.91, 138.29, 141.74, 145.29
Mobile Developer - Mid., 87.81, 90.01, 92.25, 94.56, 96.93
Mobile Developer - Sr., 129.89, 133.14, 136.46, 139.88, 143.38
Release and Configuration Manager - Mid., 99.26, 101.74, 104.28, 106.89, 109.56
Senior Analytics Consultant, 201.91, 206.96, 212.14, 217.44, 222.88
Senior Technical Writer/Editor, 116.69, 119.61, 122.60, 125.66, 128.81
Software Application Architect, 137.93, 141.38, 144.92, 148.54, 152.26
Sr. Software Engineer, 170.43, 174.69, 179.05, 183.53, 188.11
Lead Analyst, 135.42, 138.80, 142.27, 145.82, 149.47
Senior Project Director, 226.46, 232.12, 237.92, 243.87, 249.96
Senior Architect, 159.60, 163.59, 167.68, 171.87, 176.17
```

```
  File "<ipython-input-2-2c564b6586a1>", line 2
    Computer Specialist I, 66.53, 68.19, 69.89, 71.64, 73.43
                   ^
```

In [13]:
```
# Load in created CSV with no column headers
icf = pd.read_csv('rate_schedules/_initial_lists/icf_price_list.csv', header=None)

# Name columns based on GSA data
icf.rename(columns={0:'05/27/16 - 05/26/17',↔

# Log jobs based on on GSA data
role = ['Principal Systems Analyst', 'Senior Systems Analyst', 'Systems Analyst',
        'Principal Programmer', 'Senior Programmer', 'Programmer', 'Junior Programmer',
        'Principal Software Engineer', 'Senior Software Engineer', 'Software Engineer',
        'Junior Software Engineer', 'Senior Network Engineer', 'Network Engineer',
        'Principal Data Specialist/Informaticist', 'Senior Data Specialist/Informaticist',
        'Data Specialist/Informaticist', 'Junior Data Specialist/Informaticist',
        'Senior Project Manager', 'Project Manager', 'Task Manager', 'Senior Creative Designer',
        'Technical Writer', 'Creative Designer', 'Junior Creative Designer',
        'Principal Cyber Security Specialist', 'Senior Cyber Security Specialist',
        'Cyber Security Specialist', 'Junior Cyber Security Specialist',
        'Senior User Experience Specialist', 'User Experience Specialist',
        'Junior User Experience Specialist', 'Expert Technologist III', 'Expert Technologist II',
        'Expert Technologist I', 'Associate I', 'Analyst III', 'Analyst II', 'Analyst I',
        'User Support/Clerical II', 'User Support/Clerical I', 'Subject Matter Expert IV',
        'Subject Matter Expert III', 'Subject Matter Expert II', 'Subject Matter Expert I']

# Assign roles to dataframe column and rearrange columns
icf['Role'] = role
icf = icf[['Role','05/27/16 - 05/26/17','05/27/17 - 05/26/18',
           '05/27/18 - 05/26/19','05/27/19 - 05/26/20','05/27/20 - 05/26/21']]

# Log education information based on GSA data
education = ['Bachelor's Degree', 'Bachelor's Degree', 'Bachelor's Degree', 'Bachelor's Degree',↔

# Log responsibilities based on GSA data
responsibility = ["Develop and modify systems and subsystems. Support gathering information from users, defini

# Log required years of experience based on GSA data
yoe = [8,6,3,8,6,3,0,8,6,3,0,6,3,8,6,3,0,8,6,4,6,4,3,0,8,6,3,0,6,3,0,12,10,8,3,2,1,0,0,0,15,10,8,6]

# Assign logged information to newly created columns
icf['Education'] = education
icf['Functional Responsibility'] = responsibility
icf['Years of Experience'] = yoe
icf['Company'] = 'ICF Incorporated'
icf['socio_economic'] = 'Other than small business'

# Save dataframe to a new csv, observe and inspect to make sure orders are correct
icf.to_csv('rate_schedules/icf_rates.csv')
icf
```

Basic Wrangling & Cleaning

In order to create a uniform set of datasets, we had to standardize some of the data.

- We converted various timelines to specific year-to-year dates.
- Differentiated each company within each dataset.
- Standardized column names and csv formats/layouts.
- Filled all unfilled elements with a nan (not a number).
- Performed basic statistical analysis to add to each dataset.
- Merge all data into a master dataframe.

	Role	2018_2019	2019_2020	2020_2021	2021_2022	2022_2023	2023_2024	2024_2025	Education	Functional Responsibility	Years of Experience	Company	socio_economi
0	Senior Program Manager	442.04	454.86	468.05	481.62	NaN	NaN	NaN	Bachelor's	The Senior Program Manager has overall account...	15	Accenture Federal Services	Other than smal business
1	Program Manager	389.42	400.71	412.33	424.29	NaN	NaN	NaN	Bachelor's	Program Managers plan and manage projects to c...	12	Accenture Federal Services	Other than smal business
2	Project Manager	252.85	260.18	267.73	275.49	NaN	NaN	NaN	Bachelor's	The Project Manager manages, plans and coordin...	10	Accenture Federal Services	Other than smal business
3	Task Manager	190.81	196.34	202.03	207.89	NaN	NaN	NaN	Bachelor's	Task Managers apply their broad management ski...	7	Accenture Federal Services	Other than smal business
4	Subject Matter Expert 1	225.85	232.40	239.14	246.08	NaN	NaN	NaN	Bachelor's	The Subject Matter Expert 1 has industry exper...	10	Accenture Federal Services	Other than smal business

Initial Trial & Error

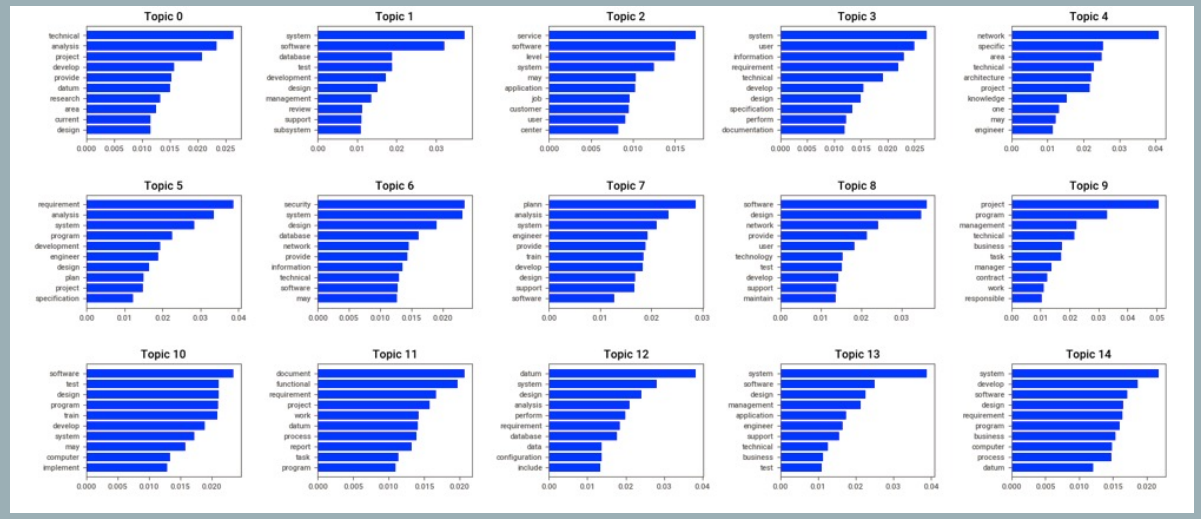
Initially we tried using another set of methods. We attempted Cosine Similarity & Topic Modeling. Here are the basic premises:

- Cosine Similarity – uses cosine algorithms to score phrases based on similar words and frequency

  I love dogs **is more closely related to** *I love cats* **as opposed to** *I love love love dogs* ***because the frequency of love will score the phrases lower, even though dog is in both***

- Topic Modeling – creates clusters based off of the amount of similar words within a sentence or phrase

This did not work because neither model could look at the order of the words as sentences. They basically were both word counters, rather than phrase readers, which doesn't help when trying to compare role responsibilities.



Preprocessing

In the preprocessing phase, we set up the responsibilities (phrases) so that they may be correctly read within NLP. In order to do this many factors were taken:

- Lowercase all values
- Remove special characters
- Remove 'ing'
- Remove numbers
- Remove double spaces
- Strip outside spaces
- Decontract all words (won't ---> will not)
- Remove stopwords (the most common words in English language)
- Lemmatize words (kept ---> keep, are ---> be)

This makes it so that all variations of a word will be limited; a computer can't read uppercase and lowercase and distinguish it's the same word. Also stopwords add no value to a score, so they just add unneeded processing time during NLP.

```
# Try and think of ways to expand contractions
def decontracted(phrase):
    # specific
    phrase = re.sub(r"won\'t", "will not", phrase)
    phrase = re.sub(r"can\'t", "can not", phrase)

    # general
    phrase = re.sub(r"n\'t", " not", phrase)
    phrase = re.sub(r"\'re", " are", phrase)
    phrase = re.sub(r"\'s", " is", phrase)
    phrase = re.sub(r"\'d", " would", phrase)
    phrase = re.sub(r"\'ll", " will", phrase)
    phrase = re.sub(r"\'t", " not", phrase)
    phrase = re.sub(r"\'ve", " have", phrase)
    phrase = re.sub(r"\'m", " am", phrase)
    return phrase

# Apply contraction
contracted = []
for row in df.responsibility:
    contracted.append(decontracted(row))

# reassign contraction
df['responsibility'] = contracted
df.head(1)
```

Variations of String Similarity

1. Edit distance based: algorithms falling under this category try to compute the number of operations needed to transforms one string to another. More the number of operations, less is the similarity between the two strings. One point to note, in this case, every index character of the string is given equal importance.

2. Token-based: in this category, the expected input is a set of tokens, rather than complete strings. The idea is to find the similar tokens in both sets. More the number of common tokens, more is the similarity between the sets. A string can be transformed into sets by splitting using a delimiter. This way, we can transform a sentence into tokens of words or n-grams characters. Note, here tokens of different length have equal importance.

3. Sequence-based: here, the similarity is a factor of common sub-strings between the two strings. The algorithms, try to find the longest sequence which is present in both strings, the more of these sequences found, higher is the similarity score. Note, here combination of characters of same length have equal importance.

In this case, we use a combination of token-based and sequence-based string similarity to score based off the amount of similar words (token) and how long each substring lasts before breaking (sequence). This combination helps us create something along the lines of an actual phrase being compared to one another.

Variations of String Similarity

1

```
>> import textdistance
>> textdistance.hamming('text', 'test')
1
>> textdistance.hamming.normalized_similarity('text', 'test')
0.75
>> textdistance.hamming('arrow', 'arow')
3
>> textdistance.hamming.normalized_similarity('arrow', 'arow')
0.4
```

2

```
>> tokens_1 = "hello world".split()
>> tokens_2 = "world hello".split()
>> textdistance.sorensen(tokens_1 , tokens_2)
1.0
>> tokens_1 = "hello new world".split()
>> tokens_2 = "hello world".split()
>> textdistance.sorensen(tokens_1 , tokens_2)
0.8
```

3

```
>> string1, string2 = "i am going home", "gone home"
>> textdistance.ratcliff_obershelp(string1, string2)
0.66
>> string1, string2 = "helloworld", "worldhello"
>> textdistance.ratcliff_obershelp(string1, string2)
0.5
>> string1, string2 = "test", "text"
>> textdistance.ratcliff_obershelp(string1, string2)
0.75
>> string1, string2 = "mes", "simes"
>> textdistance.ratcliff_obershelp(string1, string2)
0.75
>> string1, string2 = "mes", "simes"
>> textdistance.ratcliff_obershelp(string1, string2)
0.75
>> string1, string2 = "arrow", "arow"
>> textdistance.ratcliff_obershelp(string1, string2)
0.88
```

Final Output & Results

For the final output (dataset/dataframe), we:

- Perform the string similarity algorithms on the preprocessed phrases (role responsibilities)
- Receive a score
- Align the comparing role, responsibility and contractor to every other variation of competitors
- Find the average of the scores
- View visual of which roles stand out
- Make a decision to lower our rate or give a new hire an offer based off of competitor rates in the responsibility

```
In [32]: ss[(ss.gunnison_role == 'Computer Specialist I') & (ss.comparing_role == 'Administrative Assistant')]
```

Out[32]:

	gunnison_role	comparing_role	comparing_company	gunnison_func_resp	comparing_func_resp	gunnison_processed_resp	comparing_processed_resp	tb_jaccard_index	tb_sorensen_coefficient	tb_cosine_similarity	sb_ratcliff_obershelp_similarity	average
78	Computer Specialist I	Administrative Assistant	21st Century Solutions	Requires technical research and writing skills...	Assists and provides executive support by coor...	require technical research writ skill provide ...	assist provide executive support coordinat off...	0.069767	0.130435	0.130931	0.335106	0.166560
170	Computer Specialist I	Administrative Assistant	Cyberlink	Requires technical research and writing skills...	Technical administrative work such as uploadin...	require technical research writ skill provide ...	technical administrative work upload file docu...	0.051282	0.097561	0.100000	0.286604	0.133862
555	Computer Specialist I	Administrative Assistant	PSI International	Requires technical research and writing skills...	Use PCs to maintain and update office records....	require technical research writ skill provide ...	use pc maintain update office record aid staff...	0.028169	0.054795	0.057735	0.262877	0.100894
647	Computer Specialist I	Administrative Assistant	21st Century Solutions	Requires technical research and writing skills...	Assists and provides executive support by coor...	require technical research writ skill provide ...	assist provide executive support coordinat off...	0.069767	0.130435	0.130931	0.335106	0.166560
739	Computer Specialist I	Administrative Assistant	Cyberlink	Requires technical research and writing skills...	Technical administrative work such as uploadin...	require technical research writ skill provide ...	technical administrative work upload file docu...	0.051282	0.097561	0.100000	0.286604	0.133862
1124	Computer Specialist I	Administrative Assistant	PSI International	Requires technical research and writing skills...	Use PCs to maintain and update office records....	require technical research writ skill provide ...	use pc maintain update office record aid staff...	0.028169	0.054795	0.057735	0.262877	0.100894

Final Output & Results

We want normal distribution to see that our scoring method is actually working statistically, and we want outliers. The outliers are the very few roles that have very similar responsibilities to the role we are comparing them too.