Programming in Big Data
Final Project

Profile Suitability Estimator

Team Members:
Mayank Lamba
(ml5711)

Anshul Sharma
(as10950)

Sahil Goel
(sg5591)

Abstract

With increasing number of job postings and people applying for them it is becoming difficult for the recruiters to assess all resumes carefully and match them with suitable job descriptions due to which many good candidates goes unnoticed. On an average a recruiter spends around 7.4 seconds to look at a candidate's resume and is not a good way to assess their skills. A good match between job descriptions and a candidate's resume is very important both for the candidate and recruiter.

Motivation

Though there are many Applicant Tracking Systems like Jobvite, Taleo, etc. in the market, but the inclination towards applying machine learning techniques in an innovative way is still a distant goal. There is a requirement of a system that can analyze the huge amounts of data in the job application sphere to not only find valuable insights but also provide a utility that can improve, both in terms of speed and accuracy, the existing systems for matching resumes with job descriptions.

Introduction

The main objective of the project is to build a scalable pipeline to compare job resumes and job descriptions using the power of Big Data and Spark. The final system that we develop will be able to suitably compare all candidate resumes a recruiter gets for a specific job description and provide a similarity score to each resume with respect to the job description it is being compared to. These similarity scores will help a recruiter pick only the top resumes as per their need and will help in saving their time and effort. Additionally, the system we develop would analyze the text present in job descriptions and resumes and generate insights about the skills, tools and kind of work experience needed for a job posting and can help the candidates understand the requirements of the job market.

Data Description

Data for this project was accumulated from various sources including kaggle, github and google. Following are the major components of our dataset:

1. **Job Description Dataset**: It is a culmination of various job descriptions of domains including IT, Legal, Management, Engineering, Food industry etc. fetched from a github repository.
2. **Resume Dataset**: These are the resumes (of various candidates) collected in the form of pdfs, docx files and text from various csv files found on sources like github, kaggle and other data sources. This complete dataset consists of both fresh college graduates as well as experienced professionals working as Data Scientist, Software Engineers, Consultants, Store Managers, Trading Analysts etc.
3. **Customized Lexicon**: These are customized csv files containing Technical and Non-technical skills and are used during the exploratory analysis of the text.

Technologies Used:

We have used the following technologies for our data preparation, analysis and evaluation:

- Apache Spark 2.1.0 : We used the *pyspark* interpreter in the zeppelin notebook for writing the entirety of our code. It was used so that we could clean and transform our data in a way that could also be done at scale. No matter how huge the data gets, as spark can work on distributed systems, this code can use those capabilities and handle that data.
- Apache Zeppelin 0.7.2 : We used the zeppelin notebook as a medium to write our code using the *pyspark* interpreter.
- Python Matplotlib library for visualizations
- Pyspark ML libraries for creating the pipelines for implementation of the word embeddings.

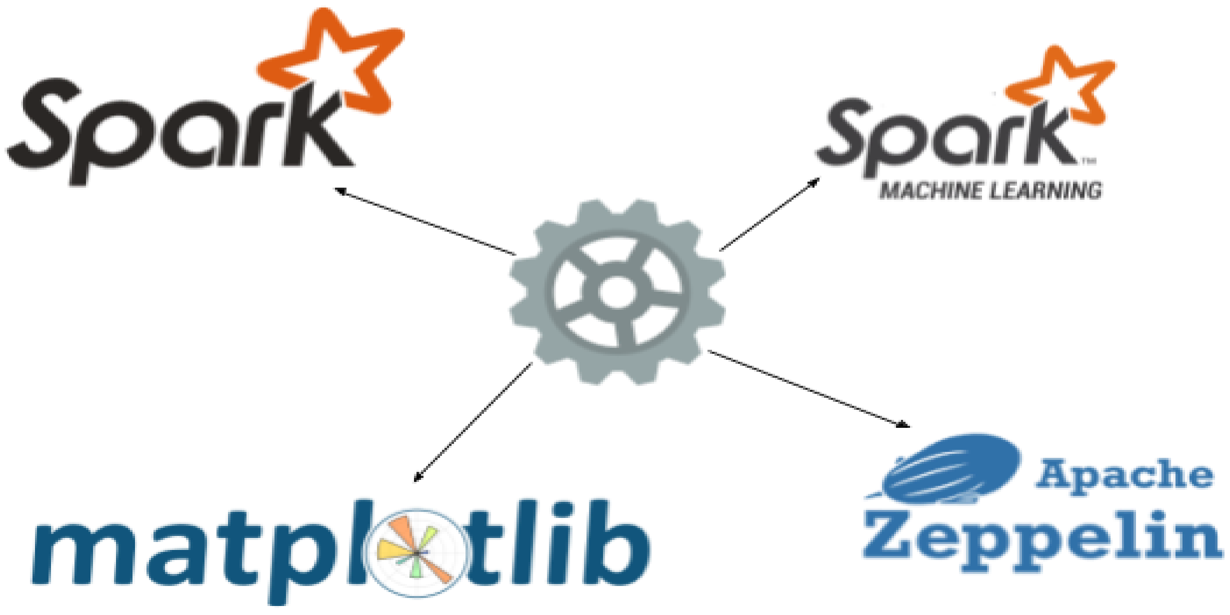


Figure 1

Spark Pipeline:

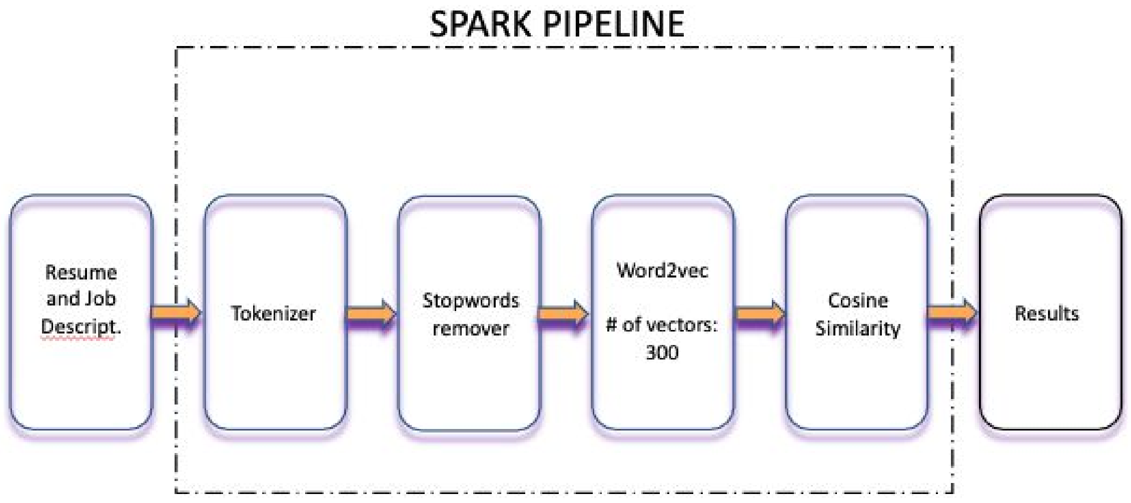


Figure 2

Tokenizer:

Word tokenization is the process of taking text (such as a sentence) and breaking it into individual terms (usually words)

Stopwords remover:

Stop words are words which should be excluded from the input, typically because the words appear frequently and don't carry as much meaning.

Word2vec:

Word2vec is a two layer neural network that transforms text into word embeddings. The purpose and usefulness of Word2vec is to group the vectors of similar words together into the vector space. The implementation in pyspark that we used is using the Skip-gram model. In our model we convert

Cosine Similarity:

The cosine similarity between two vectors (or two documents on the Vector Space) is a measure that calculates the cosine of the angle between them. Basically word embeddings are used in vector space to calculated the cos angle between them.

Text Analysis and Exploration:

Exploring most required technical skills in the job descriptions:
In this section we have tried to analyze the technical skills required across various job descriptions. By plotting the instances of each skill across each job description we can see that **SQL** is the most demanding technical skill, followed by java and C. So, people possessing these skills will be in more demand than others.

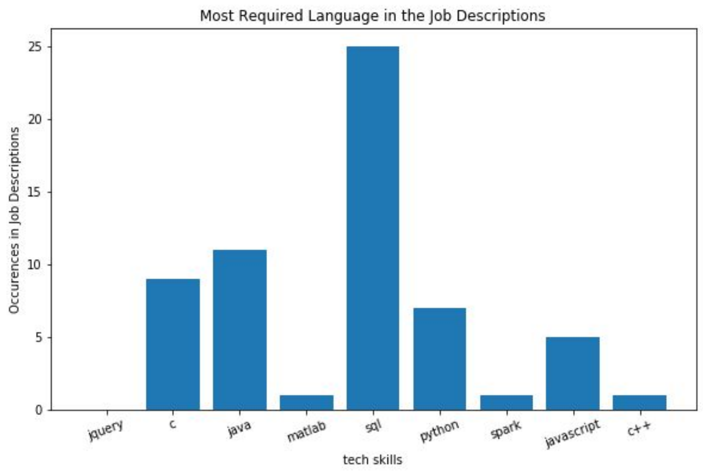


Figure 3

Exploring most required non-technical skills in the job descriptions:
After analyzing the non-technical skills present in the job descriptions, we found that **Management** followed by **Communication** are two most important skills required in non-technical jobs. So a candidate having good management and communication skills are required in the Non-tech industry.

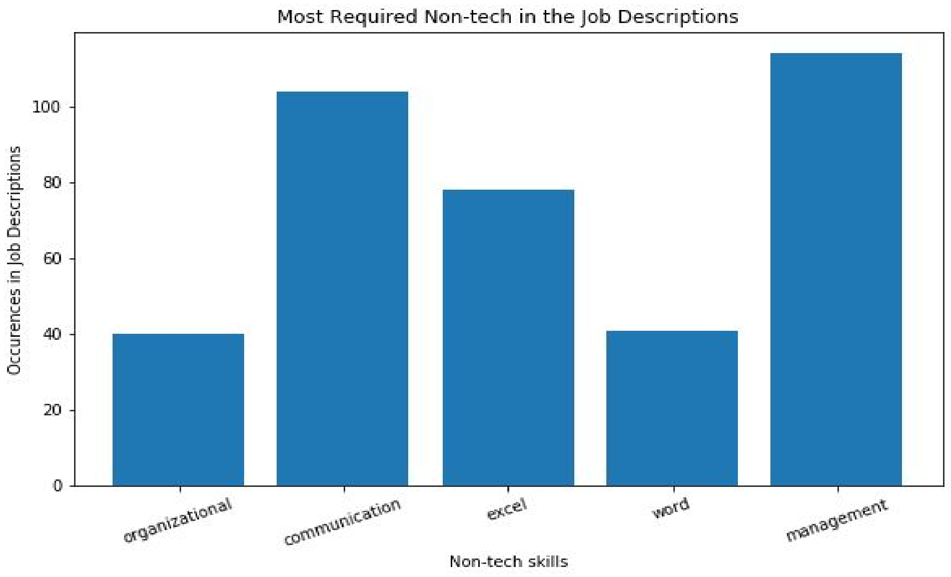


Figure 4

Exploring demanded work experience in the job descriptions:
We have plotted a bar chart by extracting the work experience required by various job descriptions. As we can see from the graph below, the most common work-experience required by various jobs is around 5 years, followed by 3 years and less. So, we can make an inference that people who have just joined the industry or have worked for few years are most sought after.

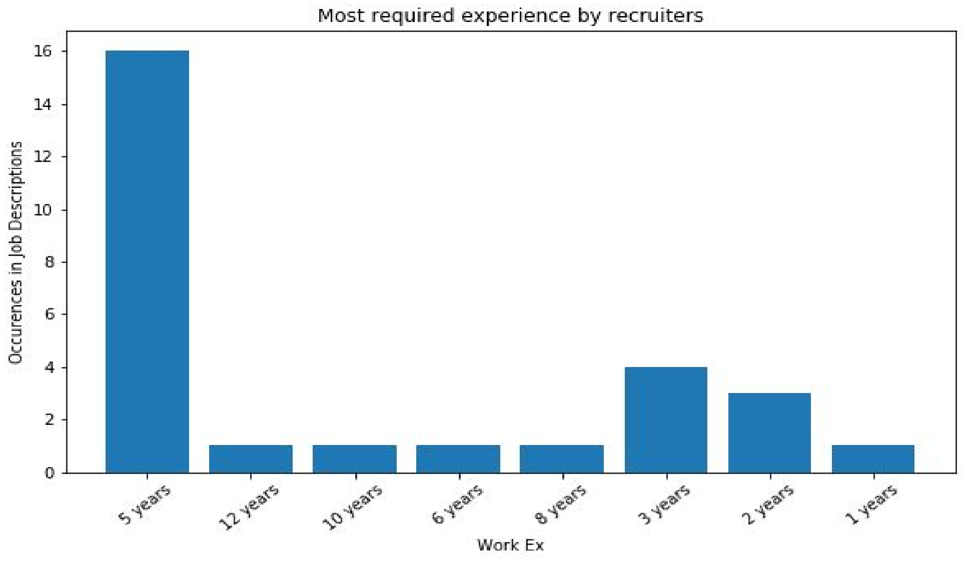
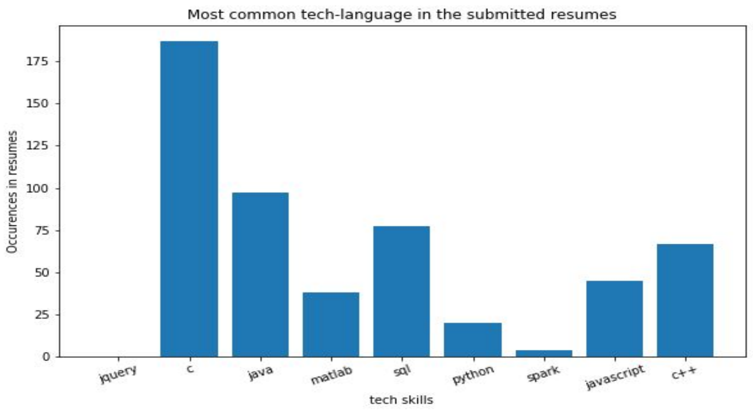


Figure 5

Exploring most prominent Tech skills in Resumes:
When we used resumes to analyze technical skills of candidates we found the most of the candidates have **C programming language** on their resume followed by **Java and Sql,** but C programming language is not the most sought skill in job descriptions. So there is a skill gap between what the recruiters are looking for and what a candidate possess in the technical sector.



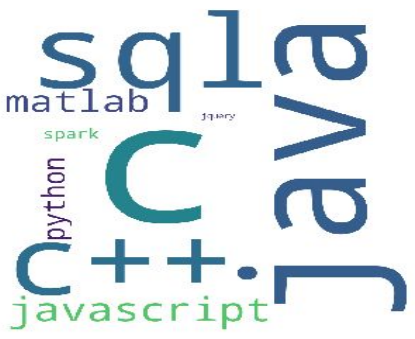


Figure 6

Exploring most prominent Non-Tech skills in Resumes:

Next, we have tried to get insights about the prominent Non-tech skills from various resumes. From the plot below, we can observe that **Management** and **Communication** are two most common skills present in resumes. Also, these two skills were most sought after in job descriptions too. So, we can infer that candidates applying for Non-tech jobs are aligned with the job descriptions in terms of skills.

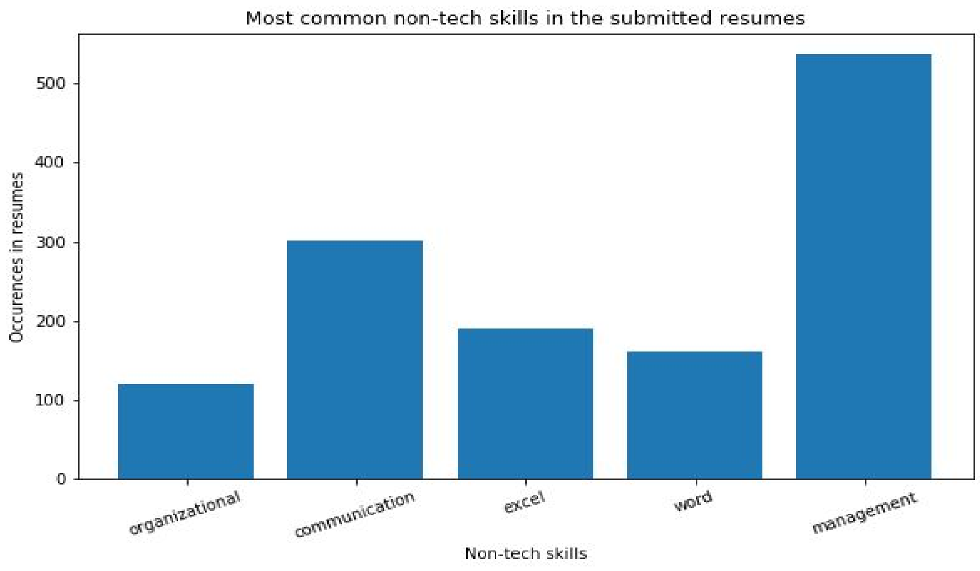
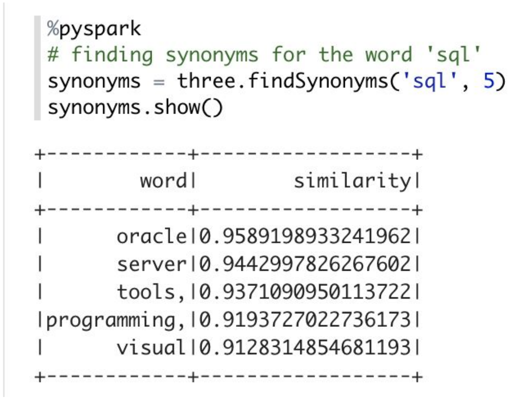


Figure 7

Results:

Finally after word to vector conversion for both job description and resumes and calculating cosine difference, we were able to get top ten resumes for each job description on the basis of similarity score. First, let us look at some results with the corpus that we trained for the Word2Vec on resume dataset:



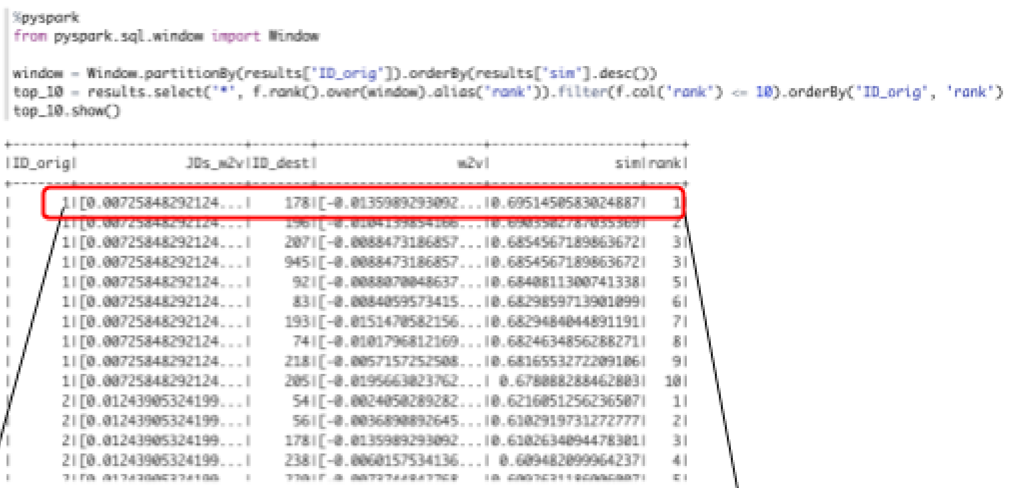
```
%pyspark
# finding synonyms for the word 'sql'
synonyms = three.findSynonyms('sql', 5)
synonyms.show()

+------------+------------------+
|        word|        similarity|
+------------+------------------+
|      oracle|0.9589198933241962|
|      server|0.9442997826267602|
|      tools,|0.9371090950113722|
|programming,|0.9193727022736173|
|      visual|0.9128314854681193|
+------------+------------------+
```

Figure 8

As can be clearly seen, synonyms for the word '*sql*' in the figure 8 represent its significance with respect to the corpus it was trained in.

The following figure 9 shows the results after applying the window function to get the top 10 resumes with respect to each job description. It is important to note that the *ID_orig* is the IDs from the Job Descriptions table and *ID_dest* has the IDs from the resume dataset. Let us look at the topmost match for the first job description. It matches with the 178th resume with a similarity score of 0.69. It also shows the text from both job description and resume with the respective Ids which as can be seen are decently similar.



```
%pyspark
from pyspark.sql.window import Window

window = Window.partitionBy(results['ID_orig']).orderBy(results['sim'].desc())
top_10 = results.select('*', f.rank().over(window).alias('rank')).filter(f.col('rank') <= 10).orderBy('ID_orig', 'rank')
top_10.show()
```

```
+-------+--------------------+-------+--------------------+------------------+----+
|ID_orig|             JDs_m2v|ID_dest|                 m2v|               sim|rank|
+-------+--------------------+-------+--------------------+------------------+----+
|      1|[0.00725848292124...|    178|[-0.0135989293092...|0.6951450583024887|   1|
|      1|[0.00725848292124...|    196|[-0.0104139854186...|0.6903502787055369|   2|
|      1|[0.00725848292124...|    207|[-0.0088473186857...|0.6854567189863672|   3|
|      1|[0.00725848292124...|    945|[-0.0088473186857...|0.6854567189863672|   3|
|      1|[0.00725848292124...|     92|[-0.0088070048637...|0.6840811300741338|   5|
|      1|[0.00725848292124...|     83|[-0.0084059573415...|0.6829859713901099|   6|
|      1|[0.00725848292124...|    193|[-0.0151470582156...|0.6829484044891191|   7|
|      1|[0.00725848292124...|     74|[-0.0101796812169...|0.6824634856288271|   8|
|      1|[0.00725848292124...|    218|[-0.0057157252508...|0.6816553272209106|   9|
|      1|[0.00725848292124...|    205|[-0.0195663023762...| 0.678088288462803|  10|
|      2|[0.01243905324199...|     54|[-0.0024050289282...|0.6216051256236507|   1|
|      2|[0.01243905324199...|     56|[-0.0036890892645...|0.6102919731272777|   2|
|      2|[0.01243905324199...|    178|[-0.0135989293092...|0.6102634094478301|   3|
|      2|[0.01243905324199...|    238|[-0.0060157534136...| 0.609482099964237|   4|
|      2|[0.01243905324199...|    [illegible]|[illegible]|[illegible]|   5|
```

Some text from the Job Description w.r.t ID:

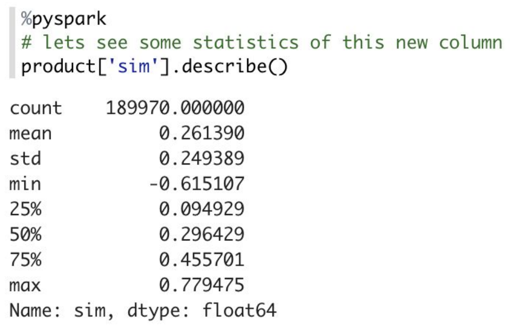
Graduate Interns with strong expertise/passion in BI or Big Data & Analytics Solutions (SAP BW or SAP HANA or Oracle / MS SQL EDW / PL/SQL / BODS / SAS / Big Data / Visualization Tools) to join our College Recruiting Hiring Program

Some text from the Resume w.r.t ID:

Skills\nLanguages: C#, Java, C, C++, PHP\nFrameworks: .NET, Swing, Qt\nWeb Development: HTML, CSS, JavaScript
Databases: Oracle, SQL, PostgreSQL

Figure 9

Following are some more statistics w.r.t the similarities column(‘*sim*’):



```
%pyspark
# lets see some statistics of this new column
product['sim'].describe()
```

```
count    189970.000000
mean          0.261390
std           0.249389
min          -0.615107
25%           0.094929
50%           0.296429
75%           0.455701
max           0.779475
Name: sim, dtype: float64
```

Figure 10

As we can see that the highest match was about 77% which is a decent score with respect to the data that was present.

Conclusion & Future Scope:

- We are trying to implement one hot encoding over industry fields to add relevant dimensions to the vectors and improve similarity scores.
- We would like to implement PCA over the vectors to reduce dimensionality.
- Also, we can aggregate data from social media platforms like facebook and linkedin to suitably rank the candidate.

In conclusion, we can see that the system that we have developed can be implemented in various applications to facilitate recruiters in a substantial way.

References:

- Google’s Word2vec paper by Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean. ‘Distributed Representations of Words and Phrases and their Compositionality’.
- Faisal Rahutomo*, Teruaki Kitasuka, and Masayoshi Aritsugi. ‘Semantic Cosine Similarity’.
- Spark ML guide https://spark.apache.org/docs/latest/ml-guide.html
- Saket Maheshwary, Hemant Mishra. 2018. ‘Matching Resumes to Jobs via Deep Siamese Network’.