Automated Resume Screening

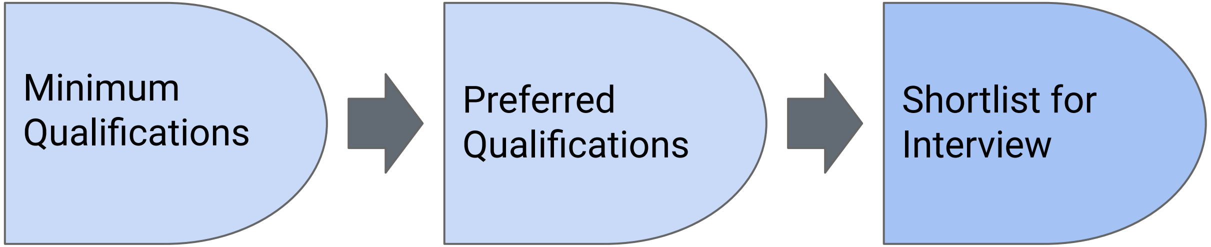
COMP 4750: Natural Language Processing

Shawon Ibn Kamal

What is Resume Screening?

Resume screening is the process of determining whether a candidate is qualified for a role based on their education, experience, and other information captured on their resume.

The usual 3 steps to resume screening:



Minimum Qualifications

Preferred Qualifications

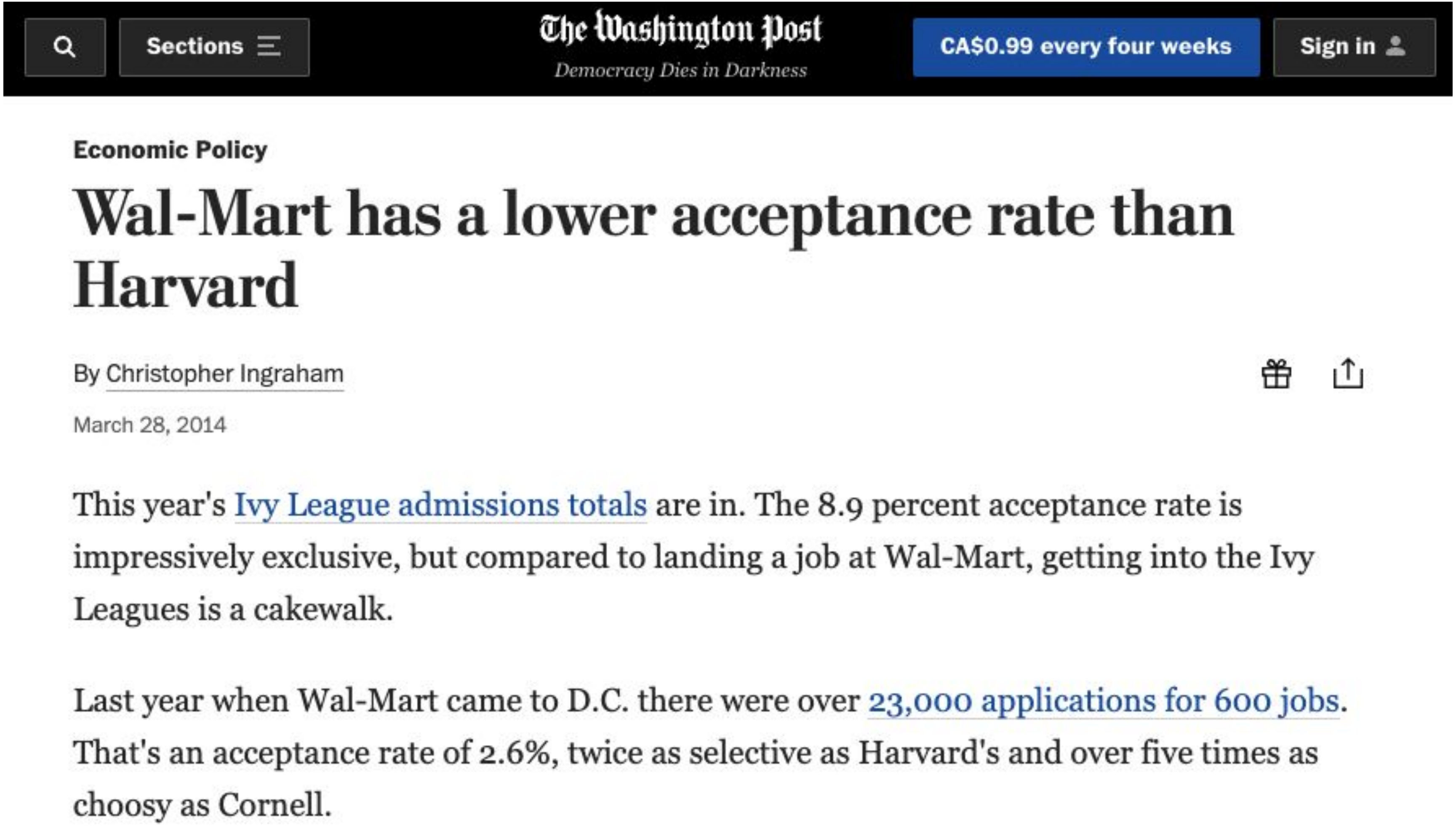
Shortlist for Interview

Challenges to screen resume manually

The biggest challenge in screening resume is in handling the **high volume** of applications received.

- On average, each corporate job offer attracts 250 resumes - upto **88%** of them tend to be **unqualified**.
- Of those candidates, 4 to 6 will get called for an interview, and only one gets the job.

Statistics: Glassdoor



Economic Policy

Wal-Mart has a lower acceptance rate than Harvard

By Christopher Ingraham

March 28, 2014

This year's Ivy League admissions totals are in. The 8.9 percent acceptance rate is impressively exclusive, but compared to landing a job at Wal-Mart, getting into the Ivy Leagues is a cakewalk.

Last year when Wal-Mart came to D.C. there were over 23,000 applications for 600 jobs. That's an acceptance rate of 2.6%, twice as selective as Harvard's and over five times as choosy as Cornell.

Challenges to screen resume manually

This high volume of resumes take up a lot of time for recruiters to make a single hire.

- Average time-to-hire a new employee was 39 days in 2016, down from 43 days in 2015.
  Source: Jobvite 2017 Recruiting Funnel Benchmark Report

More time spend on hiring implies higher cost.

- Average cost per hire statistics for companies is $4,129.
  Source: SHRM Human Capital Benchmarking Report 2016

Automated Resume Screening

Goal: The goal of this project is to built an automated resume screening application that tackles the challenges of screening manually

The application will consist of 2 main parts:

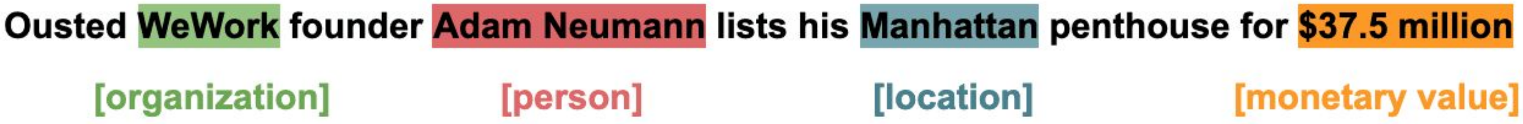
1. Parsing information from resume
2. Evaluation using ML and NLP

Parsing

- The **first** step to parsing is to extract text from PDF files. This step is quite simple. We will do it using python pdfminer.

- The **second** step is to extract specific entities from resumes: names, addresses, phone numbers, email addresses, education, experience, skills etc.
    - This will be done using a NLP technique called **Named Entity Recognition** (NER).

Parsing: Named Entity Recognition

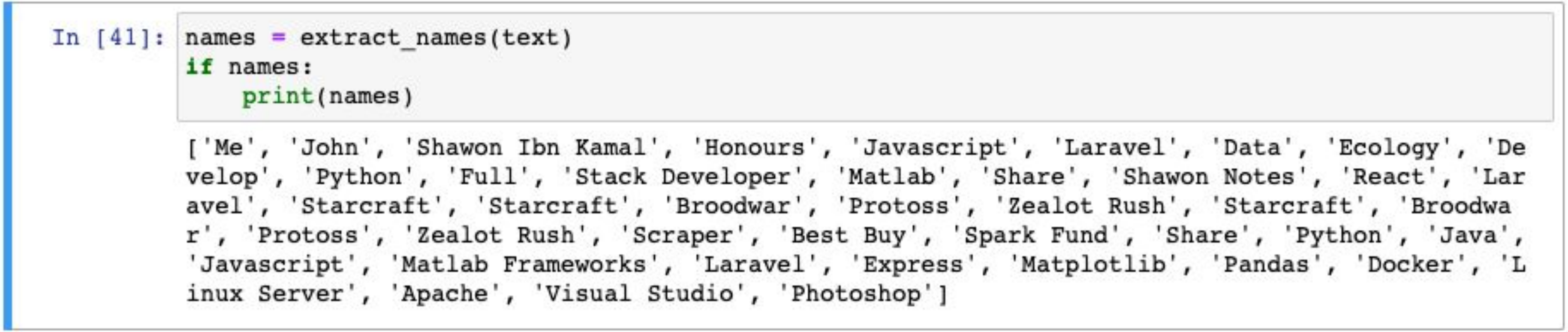


Ousted WeWork founder Adam Neumann lists his Manhattan penthouse for $37.5 million

[organization] [person] [location] [monetary value]

- A NER model will identify **noun phrase** with the help of dependency parsing and part of speech tagging (sometimes using CFG)
- Then, it will **classify** the extracted noun phrase into respective categories using lookup tables and dictionaries
- To avoid misclassification, we can create a validation layer on top. Using existing knowledge graphs can help get discrete result

Parsing: Difficulties



In [41]:

```
names = extract_names(text)
if names:
    print(names)
```

```
['Me', 'John', 'Shawon Ibn Kamal', 'Honours', 'Javascript', 'Laravel', 'Data', 'Ecology', 'De
velop', 'Python', 'Full', 'Stack Developer', 'Matlab', 'Share', 'Shawon Notes', 'React', 'Lar
avel', 'Starcraft', 'Starcraft', 'Broodwar', 'Protoss', 'Zealot Rush', 'Starcraft', 'Broodwa
r', 'Protoss', 'Zealot Rush', 'Scraper', 'Best Buy', 'Spark Fund', 'Share', 'Python', 'Java',
'Javascript', 'Matlab Frameworks', 'Laravel', 'Express', 'Matplotlib', 'Pandas', 'Docker', 'L
inux Server', 'Apache', 'Visual Studio', 'Photoshop']
```

Evaluation

In this part of the application, we need to **rank** the resumes by its relevance to the job description or requirements. It can be done using the ML process called **Information Retrieval**.

There are many ways to rank documents, here we will use an unsupervised method using **vector space model**.

The vectors representing the resume or query can be created using word2vec, Bag of Words, Inverse Document Frequency etc. In this example, we will use **Inverse Document Frequency**.

Evaluation

TF-IDF (term frequency-inverse document frequency) is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics:

- how many times a word appears in a document, and
- the inverse document frequency of the word across a set of documents.

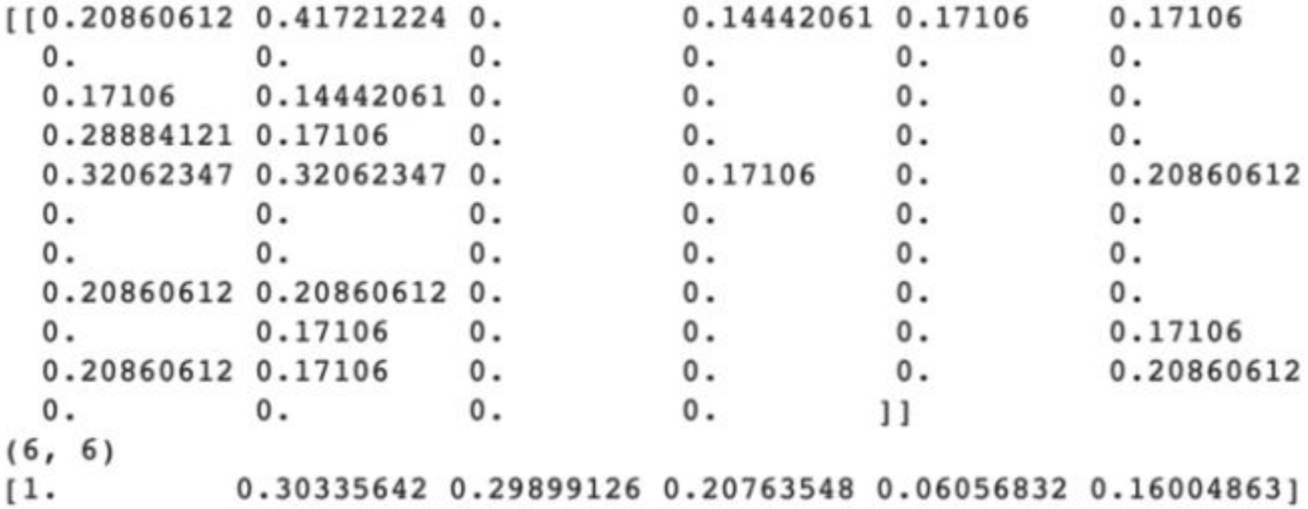
Evaluation

The vector space model will be based on **similarity**. Using cosine similarity, we can calculate a similarity score for each resumes which is given below:

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$

A -> vector of resume (document)
B -> vector of job description (query)

Evaluation



```
[[0.20860612 0.41721224 0.         0.14442061 0.17106    0.17106
  0.         0.         0.         0.         0.         0.
  0.17106    0.14442061 0.         0.         0.         0.
  0.28884121 0.17106    0.         0.         0.         0.
  0.32062347 0.32062347 0.         0.17106    0.         0.20860612
  0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.20860612 0.20860612 0.         0.         0.         0.
  0.         0.17106    0.         0.         0.         0.17106
  0.20860612 0.17106    0.         0.         0.         0.20860612
  0.         0.         0.         0.        ]]
(6, 6)
[1.         0.30335642 0.29899126 0.20763548 0.06056832 0.16004863]
```

Evaluation: Other factors

Other factors to be considered while evaluating:

- Place of location
- Years of experience
- Relevance of previous experience