Job Recommendation Engine

Harshendra

Mahak Gambhir

Vishal Gupta

IIIT Hyderabad

Introduction

- Most job applications are through resumes
- Short Listing resumes is a painful task
- Especially when there are thousands of them
- Manual checking is not feasible

Semantic Search

- What is not a semantic search ?
- Search based on keywords ?

  Ex: grep “icpc”

- Can we make it better ?
  - Weightage to sections
  - To be able to search in a particular section
  - Recommend resumes similar to a particular resume
  - NLP vs Natural Language Processing

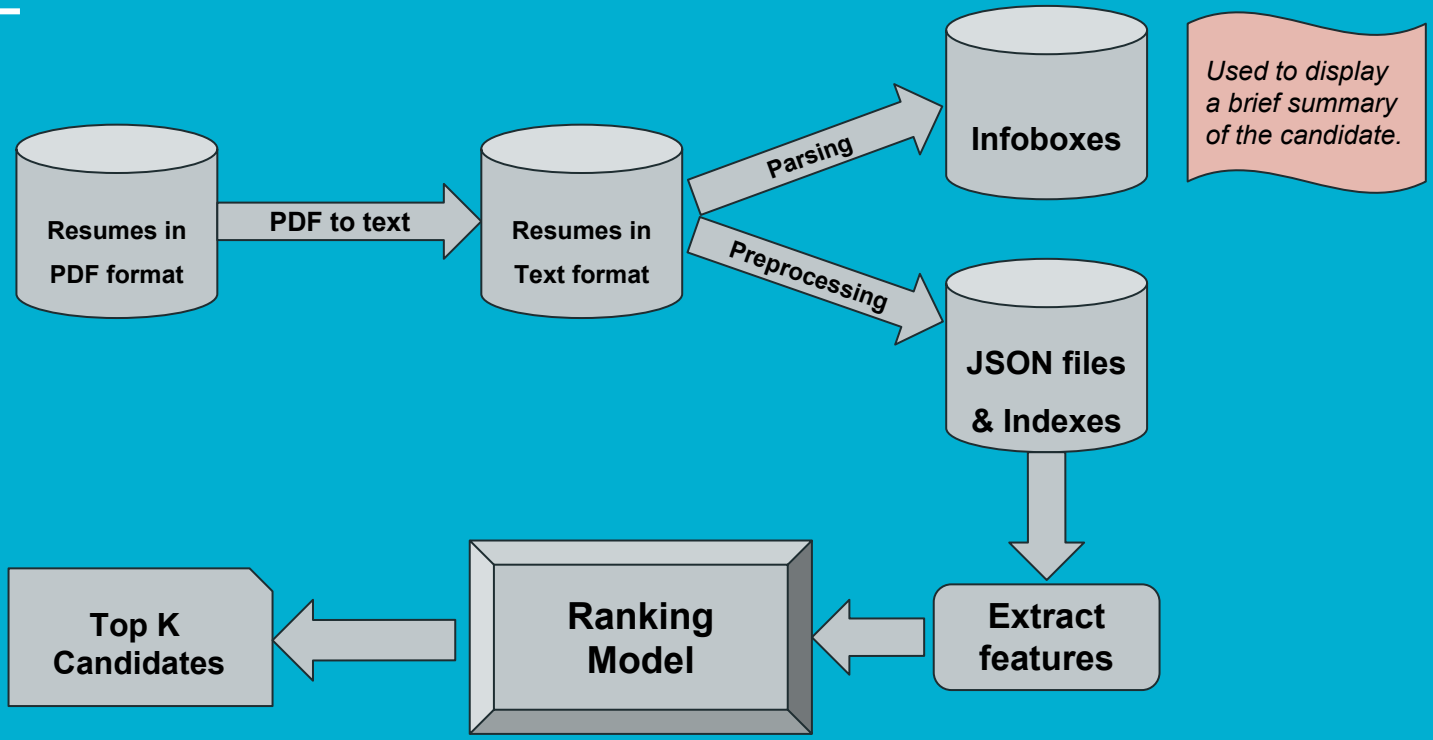
Challenges

- Structure of a resume
- Content Segregation - How well is it written ?
- Extraction of entities
  - Names
  - Marks/Grades
  - Previous Organizations
  - Years of experience
  - Skills
  - Projects
- Ranking among the shortlisted resumes

Problems addressed

- Creation of info box for each resume.
  - Name: XYZ
  - Email : abc@gmail.com
  - Phone: xx-xxxxxxxxx
  - Skills
  - Projects
- A search interface
  - Query> Machine Learning, ACM ICPC, Scikit Learn
- Recommend resumes similar to a given resume
  - Resume Path> /path/to/resume

The Process Flow



Approach

- Each resume is a text document
- Extract the sections in each resume
- NER for names
- Index the resumes according to sections
- Extract features from each resume

Technical Details

- Named Entity Recognition for extracting names
  - Used Stanford NER
- Parse the sections using most common section names
  - Project/Projects/Academic Project/Major Projects/Minor Projects
  - Skills/Technical Skills/Technical Expertise and so on
- Stanford temporal tagger for extracting the temporal information
  - Jan 2012 - Jan 2014 (2 years)
  - Useful in extracting the years of experience

Technical Details

- Bag of words approach
- Vector space model
- Extract feature for each resume
  - Boolean
    - A term is present or not d
  - Tf-Idf
    - Tf-idf weight of a term w.r.t the document d
- Rank the resumes based on distance functions
  - Cosine Similarity
  - Euclidean distance

Features

Let N be the |V|, d1 and d2 be two resumes

d_1: **v1** $<v_{11}\ v_{12}\ \cdots\ v_{1N}>$

d_2: **v2** $<v_{21}\ v_{22}\ \cdots\ v_{2N}>$

where,

v_{ij}: Tf-Idf(d_i,t_j)

t_j is the jth resume

$$\textbf{Tf-Idf(d,t)} = \mathbf{tf_t \times \log(N/df_t)}$$

tf_t - Term frequency of term t in document d

df_t - document frequency of term t

Ranking

- Convert the test query/document into tf-idf feature vector t
  Ex: test = “acm icpc, machine learning”
- Calculate cosine similarity with each document and rank them
  cos-sim(d_1,t), cos-sim(d_2,t) and so on
- Rank the resumes in non-decreasing order
- Choose top K

Tools/Frameworks

- Language: python 2.7
- Frameworks:
  - Scikit-learn
  - NItk
  - Stanford NER tagger
  - Stanford temporal tagger
- Intuition!

Demo

Links

- Code: https://github.com/gvishal/Sematic-Job-Recommendation-Engine
- Deliverables: https://www.dropbox.com/home/IRE_Major_Project/deliverables