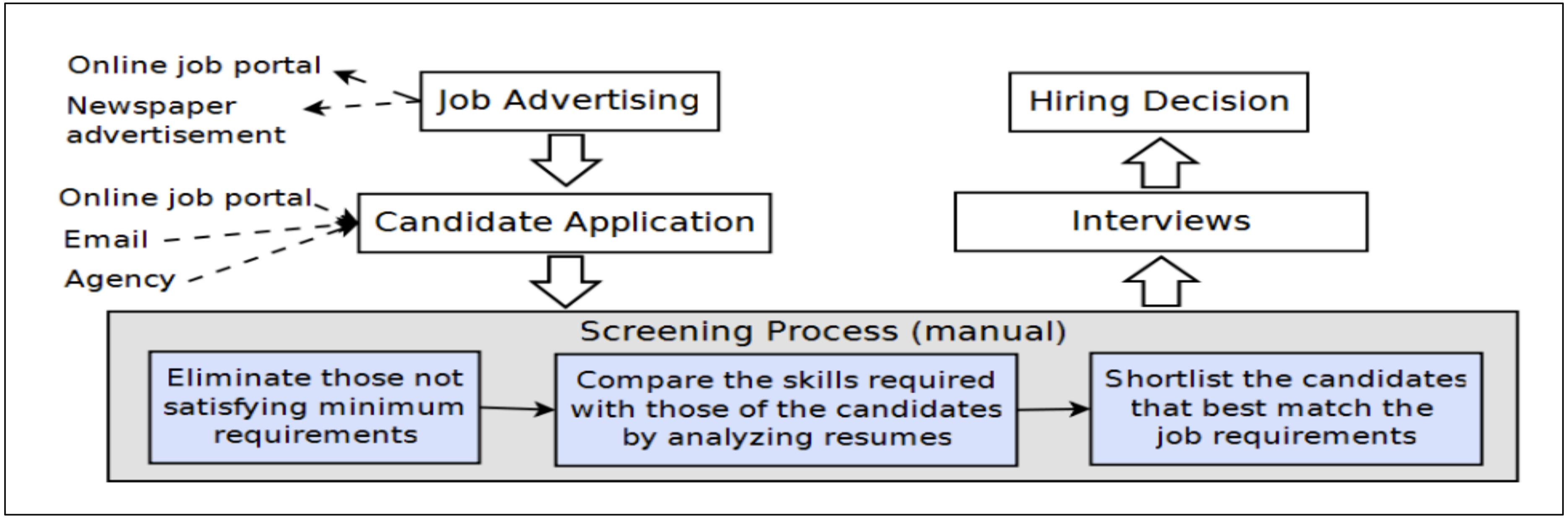
Information Extractions And Profile Recommendations Using Document Vector Embeddings And Cosine Similarity

Subhajit Das

Introduction

- One of the key challenges of the twenty first century is unemployment, a multi-dimensional social and economic phenomenon.
- Every month there is about one million people lands into the job market.
- Finding the best fitted person with appropriate skills has never been an easy feat for organizations. It requires recruiters to splash through thousands of Resumes.
- Being able to eliminate irrelevant profiles as soon as possible, could greatly help to save cost, time and efforts.

Background



Online job portal
Newspaper advertisement
Job Advertising
Hiring Decision
Online job portal
Email
Agency
Candidate Application
Interviews
Screening Process (manual)
Eliminate those not satisfying minimum requirements
Compare the skills required with those of the candidates by analyzing resumes
Shortlist the candidates that best match the job requirements

- This paper propose An AI/ML system which could tapped into this profile screening process by extracting information from semi structured resumes like skill sets, experience and accolades i.e. effective use of meta-data for profile recommendations for a given job description.

Literature Review : Information Extractions

Extracting relevant features or information from resumes and job description, mainly different types of research done earlier in this space.

- One of the first approach is to consider the keywords retrieval to screen out the resume which are not of use. This approach parses resumes for the given keyword, irrespective of what a sentence means in the resume, once the keyword is found that resume is recommended to the recruiter by this application.

- Second , published studies tried to learn the information extraction rules for resumes using XML tags to identify key attributes namely email, name, street, Province etc.

- Lastly, few studies shown information extraction is a combination of syntactical(parts of speech tagging) and semantical(topic modelling etc.)entity recognition task.

Literature Review : Vector Embeddings and Similarity

- Word vector embeddings is nothing but a vectoral representation of word in a ten to hundreds of dimensional vector space model.

- Word2Vec ,Glove Vectors and fastText are popular framework to generate word embeddings.

- Measures of distance or similarity can be computed compute between two vectors.

- Few most popular similarity measures such as : Cosine Similarity, minkowski Distance, Pearson coefficient, Jaccard similarity etc.

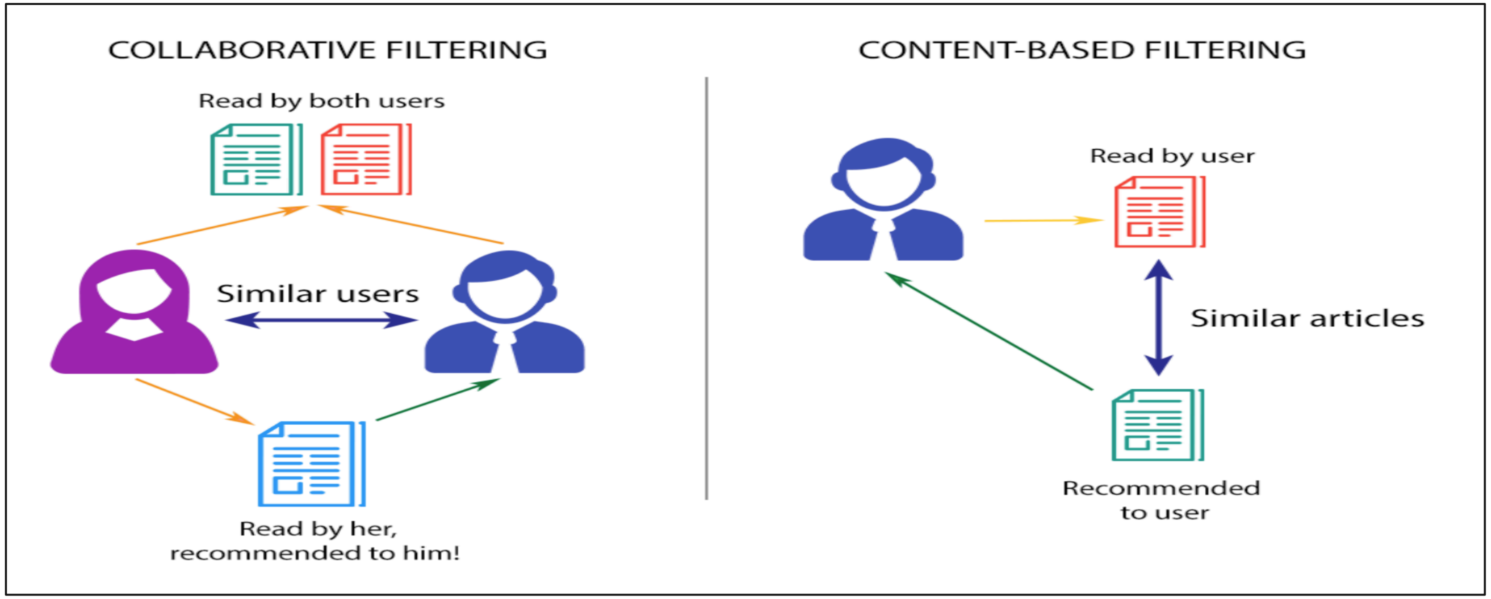
Literature Review : Recommendation Engine

- Recommender engines are the machine learned models that provide recommendations. The main objectives of recommendation engines to provide suggestions to its users base with a highly relevant set of items.

- The idea of Content Based Recommendations is to recommend items to users based on similarity between user sets and the item information

- Collaborative Filtering relies on finding people similar to the target person and recommending items which similar people have liked.

Content Based Filtering vs Collaborative Filtering



- The similarities in Collaborative Filtering technique are concerned with people’s tastes, preferences and activities, contrary to CBR where similarity is built on top of the content of the item and the user.

Research Methodology

The fundamental objective of this research paper to find the best fitted candidate’s resume from a pool of million profiles based on a particular job description. Our proposed model works in three different loosely coupled stages.

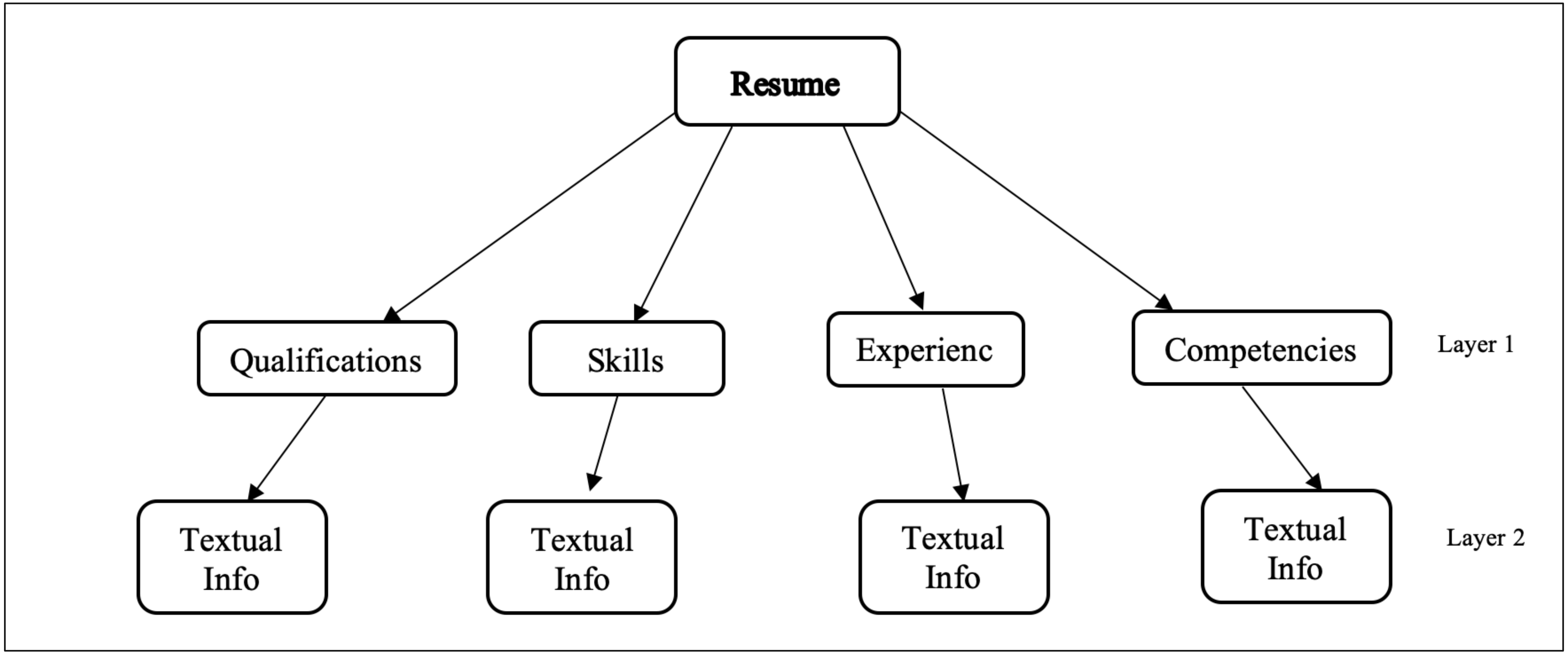
- First, it extracts information from unstructured resume convert it to structed format.
- Second, it computes the similarity between processed resume text and cleaned JD.
- Third it makes a recommendation to the recruiters.

we have done a comparative study of three different unsupervised machine learning techniques using vector embeddings and distance computing algorithms.

- TF-IDF Vector with Cosine Similarity
- K- Nearest Neighbours with Minkowski Distance
- Doc2Vec with Cosine Similarity

Research Methodology : Hierarchical Structures of Resume

A resume is mostly a semi structed data. Normally it has document level ordered hierarchical sections. Figure below illustrates the hierarchical structures of a resume.



Research Methodology : Text Processing and Information Extractions

Text pre-processing in simple terms is to bring down the raw text into its machine understandable, predictable and analysing form.

- Tokenization
- Lowercasing
- Stop Words Removal
- Lemmatization
- Noise Removal
- POS Tagging

Information Extraction(IE) pipelines start with the usual text pre-processing steps - sentence segmentation, word tokenisation and POS tagging. We thought of resumes as a composition of four sections i.e. qualification, skills, Experience and Competencies. With the help for POS tagging , NLTK and Spacy we have extracted all of those mentioned sections from the each resume and exported those into a cleaned csv file.

Research Methodology : Model Development and Recommendations

Content Based Recommendations using TF-IDF and Cosine Similarity

- TF-IDF is the most common weighting method used to describe documents in the Vector Space Model. Tf-Idf is a measure that is often used when dealing with textual data to information retrieval in particular for text mining and to calculate document similarities or to find relevant documents from a pool of documents.

K-Nearest Neighbours with Minkowski Distance

- The concept of Nearest Neighbours is to find a predefined number(k) of training samples closest in distance to the new point. The distance could be any metric like **minkowski distance**.It is a distance or similarity that measured between two points in normed vector space(N- dimensional real space) which is a generalization of the Euclidean distance and the Manhattan distance.

Research Methodology : Model Development and Recommendations

Recommendations using Doc2Vec and Cosine Similarity

- The objective of doc2vec algorithm is to generate a array of vector to represent a paragraph or document. Most of times we found paragraphs are not in its logical structures unlike words. The concept is simple but very intuitive: they have only used the word2vec model and on top of it added an extra vector i.e. Paragraph ID vector.

- Our problem is to find text similarity, so in our paper we have gone with **Content Based Recommender** where the JD given by employer and it is matched against the content of the Resumes in vector space and top N (In our study N =10) closely matching profiles are recommended to the employer. Our model is developed with the help of *Distributed Memory version of Paragraph Vector* (PV-DM).Then it computes the **cosine similarity** between every document vector of resume and particular document vector of job description and with that our proposed model recommends the top 10 most similar profile to the recruiter.

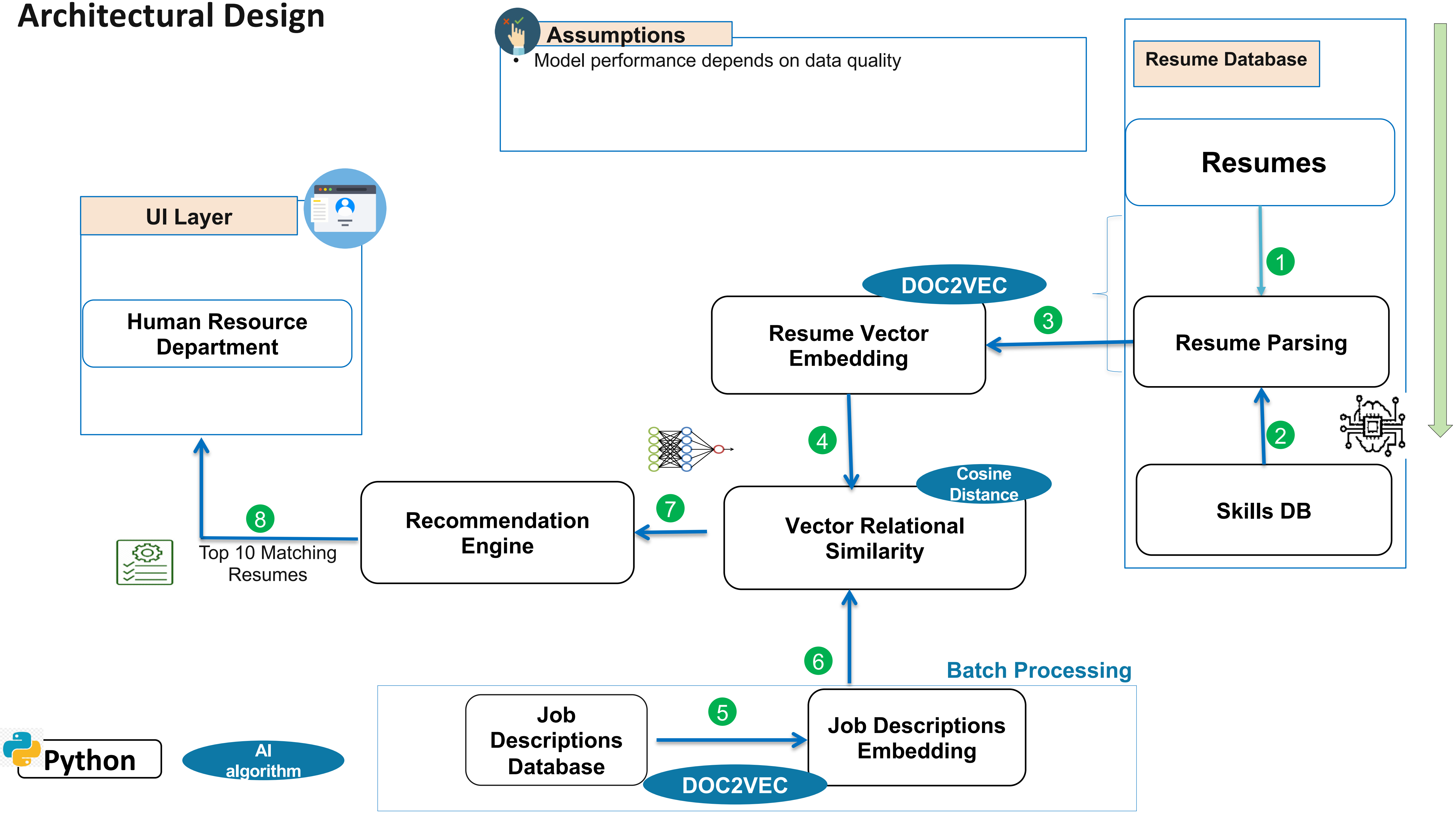
Research Methodology : Model Evaluation

- To assess our doc2vec recommendation model, we’ll first infer new vectors for each document of the training corpus, compare the inferred vectors with the training corpus, and then returning the rank of the document based on self-similarity

- Basically, greater than **91%** of the inferred documents are found to be most similar to itself and about 9% of the time it is mistakenly most similar to another document.

- Checking the inferred-vector against a training-vector is a sort of ‘sanity check’ as to whether the model is behaving in a usefully consistent manner, though not a real ‘accuracy’ value.

Architectural Design



Assumptions

- Model performance depends on data quality

Resume Database

Resumes

1

Resume Parsing

2

Skills DB

3

DOC2VEC

Resume Vector Embedding

4

Cosine Distance

Vector Relational Similarity

7

Recommendation Engine

8

Top 10 Matching Resumes

UI Layer

Human Resource Department

6

Batch Processing

Job Descriptions Database

5

DOC2VEC

Job Descriptions Embedding

Python

AI algorithm

RESULTS AND DISCUSSIONS:K- Nearest Neighbours with Minkowski Distance

To make it a comparative study between the mentioned three approaches we have consider the same JOBID to draw a clear view about our models.

Table 1 shown K- Nearest Neighbour's top 10 recommendation. We are computing similarity in nearest neighbours algorithm with the help of minkowski distance. The interpretation of the score is – the less the distance the more the similarity between JD and Resume.

JobID	Filename	Score
241	cv123.txt	1.19814596
241	cv112.txt	1.23791089
241	cv204.txt	1.2610289
241	cv142.txt	1.27997453
241	cv195.txt	1.2864011
241	cv144.txt	1.28761706
241	cv199.txt	1.28812698
241	cv140.txt	1.29116328
241	cv55.txt	1.29275112
241	cv191.txt	1.29287469

Table 1 : K-NN Top 10 Recommendation

RESULTS AND DISCUSSIONS:TF-IDF Vector with Cosine Similarity

Table 2 illustrates TF-IDF's top 10 recommended profiles. Here the similarity has been computed using cosine similarity. So the score should be interpreted like- the more the score the better matching between JD and Resume.

JobID	Filename	Score
241	cv123.txt	0.28222313
241	cv112.txt	0.23378831
241	cv204.txt	0.20490305
241	cv142.txt	0.1808326
241	cv195.txt	0.17258611
241	cv144.txt	0.17102116
241	cv199.txt	0.17036444
241	cv140.txt	0.1664487
241	cv55.txt	0.16439727
241	cv191.txt	0.16423751

Table 2 : TF-IDF Top 10 Recommendation

RESULTS AND DISCUSSIONS: Doc2Vec with Cosine Similarity

Table 3 illustrated below given an idea about content based recommendations made using doc2vec and cosine similarity. Here also the more the score the better the matching between Resume and JD.

JobID	Filename	Score
241	cv35.txt	0.9478473663330078
241	cv46.txt	0.9152319431304932
241	cv1.txt	0.8887962698936462
241	cv141.txt	0.8866349458694458
241	cv113.txt	0.8853865265846252
241	cv138.txt	0.8846480250358582
241	cv240.txt	0.8787721395492554
241	cv195.txt	0.874518871307373
241	cv127.txt	0.8744110465049744
241	cv144.txt	0.835113525390625

Table 3 : Doc2Vec Top 10 Recommendations

CONCLUSIONS AND FUTURE SCOPE

Although we have shown comparative study of three different approaches but there are lot of other ways exits. We believe this particular field of Resume recommendations based on JD has a great scope for future works, such as:

- Using other similarity metrices which have not been used in our study we could do experiments which one gives more accurate matching results.
- Collaborative Filtering and Hybrid recommendation systems could be used with some changes in data pre processing, and that will lead us to compare our study on a much boarder scope.
- Other word embeddings techniques like fastText, Glove Vectors could be used.
- Consider usage of deep learning based recommendations.
- Last but not the least a millions of resume cloud help any experiment with scalability.

THANK YOU