Searching LinkedIn using Topic Modelling

A DISSERTATION PRESENTED
BY
RAJARSHI CHOWDHURY,MOUMI DAS,SHILADITYA SWARNAKAR,SOUMYA BANERJEE,SOUVICK DATTA,SOWMYA KARTIK
TO
THE DATA SCIENCE DEPARTMENT

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
POST GRADUATE PROGRAM IN DATA SCIENCE
IN THE SUBJECT OF
CAPSTONE PROJECT

PRAXIS BUSINESS SCHOOL
KOLKATA, WEST BENGAL
APRIL 2020

Searching LinkedIn using Topic Modelling

Abstract

The purpose of the project is to reduce information asymmetry between two pools namely the recruiter pool and the candidate pool. The project uses a popular technique in natural language processing called Topic Modelling to match keywords provided by user with the relevant profiles. The model GUI takes a query as input and generates a list of pertinent profiles.

Contents

Listing of figures

The project is dedicated to our project Supervisor Professor Subhasis Dasgupta who taught us everything from scratch.

Acknowledgments

We wish to express our sincere appreciation to our supervisor, Professor Subhasis Dasgupta, who has the substance of a genius: he convincingly guided and encouraged us to be professional and do the right thing even when the road got tough. We also wish to thank our other batch-mates for their continuous help. Finally, Thank you to Praxis Business School for creating this incredible opportunity.

If opportunity doesn't knock,build a door.

-Milton Berle

1

Introduction

Organizations across the globe have always faced a tough time when it comes to acquiring the right talent with the right skill set. This becomes even more critical in the current scenario where business landscapes are changing every day and the mode of operations for business firms are evolving at a rapid rate. Another side of the story starts with the candidates who need to land on the right job. Connecting these two pools with a proper allocation is one of the central problems of Human Resource Management. Today with the advent of AI, and everything becoming automated, HR is also not lagging but the problem is far from being solved.

The recruiters' story: The recruiters difficulty lies in mapping most relevant candidates with the job requirement and with technology playing a big role in the modern world, the problems are becoming tough. The screening task is tedious and time-consuming. The screeners are under immense pressure to hire people quickly to not lose the right talent to their competitor. The rapid growth of labour intensive and knowledge-based services industry critically depends on this. Often, because of this time crunch, HR relies on small samples of the applicant pools or ranking provided by third party hiring firms for this purpose. In some cases, the best candidate for a job might have been over-looked at by a screener if things are done manually. Modern algorithms in NLP, other machine and deep learning techniques people are trying to automate the hiring process.

The candidates’ story: Another major problem is from the other side of the river, the candidates’, where to go, whom to connect with, which institutes to join to get into a good university. We often find people connecting with us on Linkedln or other social media to know about our experience and many a time it happens they just hit the wrong profile. Sad and a complete waste of time. Even on various blogs like Quora, Reddit etc. we find people asking about careers. People are clueless about various courses that are existing. We see a clear asymmetry of information. If we can connect these students with the right person in the industry or in academics both can save a lot of time and unnecessary conversations!

The problem statement: We are trying to solve is to determine a ranked list of most relevant people among a pool of profiles from a query. This can be used by recruiters to get the right candidate or a candidate to get connected to the right person and solve a little bit of the information asymmetry that is currently existing.

Our work in 50 words In our project we collected profile details of various people from LinkedIn, used natural language processing(NLP) to understand profile relevance and recommend the top 5 profiles against a query provided by the user.

Without data, you are just another person with an opinion

-W. Edwards Demings

2 Data Collection

Web pages are built using HTML and XHTML that are text-based mark-up languages. They contain a great deal of useful information in text format. However, most web pages are designed for human end-users and not for any computer program or software. To get the information from these web pages, we need something called a Scrapper- a tool to extract data from web sites.

2.1 Web Scrapping

Data displayed by websites like Facebook.com,LinkedIn.com etc. can be viewed using any web browser like Google, Mozilla, Safari etc. They do not offer the functionality to save a copy data of our interest for personal use.

Web Scraping is the automated technique employed by the scraper to extract large amounts of data from websites within a small interval of time. The data extracted can be saved to a local file in our system or a database.JSON is the common format for storage of web data and transport of that data between the client and the server. Scrappers can be either custom-built for a specific website or can be configured to work with any website(the former more predominant). Web scraping systems uses techniques like DOM parsing, computer vision and natural language processing to simulate human-like behaviour(navigation) when someone views a web-page. However, large websites use defensive mechanisms to protect their data from web scrapers and to limit the number of requests an IP may send. This makes it difficult for scraping developers to extract relevant information for the model.

2.2 Tools Used

Scrapers or spiders can be built using several techniques like Python(Beautiful Soup), Node JS, PHP, Selenium. We used the Selenium with Python API and Beautiful Soup for scrapping LinkedIn.

2.2.1 Selenium

Selenium Web Driver is a collection of open-source API. These are used to automate the testing, scraping data etc from a web application. It supports browsers like Google Chrome, Firefox, IE, Safari etc. Using the Selenium Web-Driver, we can automate the scraping process from web application pages only and it is quite easy to use. It also supports different programming languages such as C, Java, Python, etc for writing test scripts.

We have used python to write the web scraping scripts. Selenium Web-driver is independent of the platform on which it is running since the same code can be used on different Operating Systems like Windows, Linux. Selenium Web-Driver has different location strategies as well such as ID, Name, Class name, CSS selector and Xpath to extract text data.

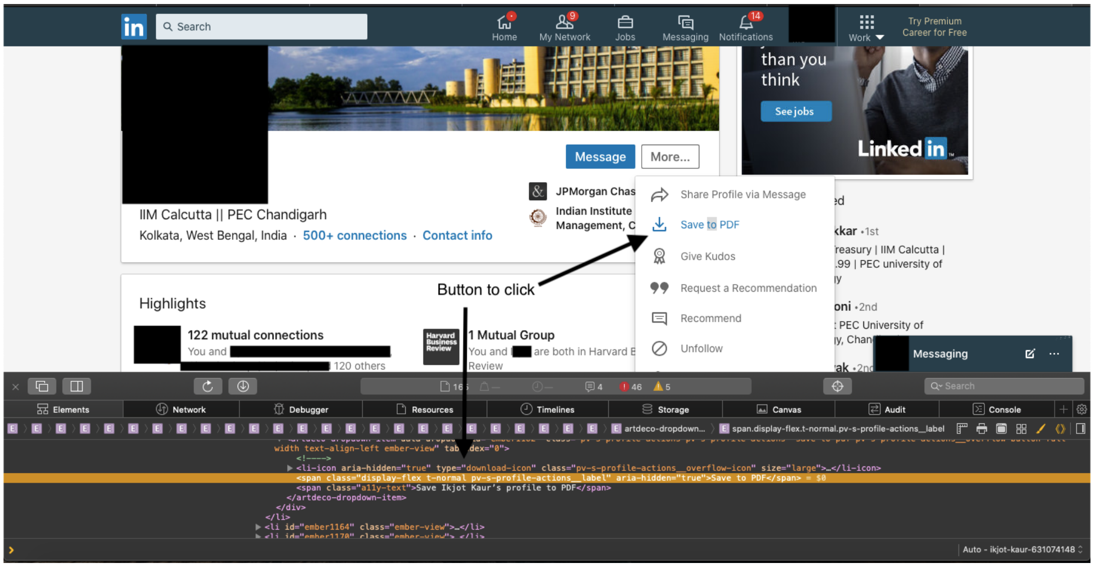
2.2.2 Beautiful Soup

Python library has Beautiful Soup (BS4) for pulling data out of web pages(HTML and XML files). It works with the parser to provide idiomatic ways to Navigate, Search and modify the parse trees. It saves programmers effort to get the relevant information. It creates a parse tree for parsed pages that can be used to extract data from HTML (including having malformed markup, i.e. non-closed tags) which is useful for web scraping.

What we did?

- Our source of data is LinkedIn.com
- We scrapped profile details using Selenium and Beautiful Soup. The first half of the data was scraped directly from the web page, the second half by downloading profile PDFs and using slate3k in Python to convert it to editable string and pre-process it(covered in next chapter).

Figure 2.2.1 shows a profile of a person being scrapped.



[redacted]

IIM Calcutta || PEC Chandigarh

Kolkata, West Bengal, India · 500+ connections · Contact info

JPMorgan Chas

Indian Institute Management, C

122 mutual connections

You and [redacted] 120 others

1 Mutual Group

You and [redacted] are both in Harvard B Review

```
width text-align-left ember-view" tabindex="0">
    <!---->
  ▶ <li-icon aria-hidden="true" type="download-icon" class="pv-s-profile-actions__overflow-icon" size="large">…</li-icon>
    <span class="display-flex t-normal pv-s-profile-actions__label" aria-hidden="true">Save to PDF</span> = $0
    <span class="a11y-text">Save Ikjot Kaur's profile to PDF</span>
  </artdeco-dropdown-item>
</div>
</li>
▶ <li id="ember1164" class="ember-view">…</li>
```

Figure 2.2.1: LinekdIn Scrapper

If you torture the data long enough, it will confess

Ronald Coase

3
Data Prepossessing

In the field of data analytics, data preparation or prepossessing of data is one of the crucial steps towards model building which plays a huge role in subsequent model building. In case of text data this becomes even more vital since the data is unstructured and needs a lot of refinement.

In linguistics, a text corpus is a large structured set of texts. Corpora are the main knowledge source in corpus linguistics. In order to make the corpora more useful for linguistic research, they are often annotated. An example of annotating a corpus is part-of-speech(POS) tagging, in which information about each word's part of speech is added to the corpus in the form of tags.Another way can be to indicate the base (lemma) form of each word. Some corpora might require further structured levels of analysis to be applied. A number of smaller corpora may be fully parsed called Parsed Corpora. The difficulty to ensure that the entire corpus is completely and consistently arises because the corpora are usually smaller, containing around one to three million words. The following prepossessing of text is done:

3.1 Cleaning Text

Cleaning of the text corpora involves removal of punctuation and unnecessary characters. This is done by string manipulation in Python.

3.2 Tokenization

For a defined document unit, tokenization is the task of cutting the document up into pieces, called tokens,simultaneously throwing away certain characters, such as punctuation.These tokens are called terms or words, but it is sometimes important to make a token distinction. They are grouped together as a semantic unit which helps in processing them.The class of all tokens containing the same character sequence is called a type. Hyphenation is used for various purposes ranging from splitting up vowels in words to joining nouns as names. It is easy to feel that the first example should be regarded as one token.Handling hyphens automatically can get complex: it can either be done as a classification problem, or more commonly by some heuristic rules.Splitting on white spaces should be regarded as a single token.

This has to be taken care of most commonly with names (John Smith), foreign phrases (au fait) and compounds that are sometimes written as a single word. Other cases with internal spaces that we need as a single token include mobile numbers and dates. Splitting tokens on spaces will give bad results. We used sklearn and NLTK's gensin package in Python to tokenize the corpus.

3.3 Lemmatization and Stemming

For grammatical reasons, documents are going to use different forms of a word, such as stabilize, stabilizes, and stabilizing. Also, there are families of derivations related words with similar meanings. In many situations, it seems as if it would be useful for a search for the root word to return documents that contain another similar word in the set.The goal of stemming and lemmatization is to reduce a word to a common base form. However, the two words differ in their approach.

Stemming usually refers to a crude process that chops off the ends of words to get the base word. This happens correctly most of the time, and often removes derivations affixes.

Lemmatization uses morphological analysis of words, to return the base or dictionary form of a word, which is known as the lemma . Porter's algorithm ([4]) is the one that has repeatedly been shown to be empirically very effective for stemming English.Lemmatizer does full morphological analysis to accurately identify the lemma for each word. Doing full morphological analysis produces best benefits for retrieval. Stemming increases recall while reducing precision.

3.4 Bag of Words

The bag-of-words model is a simplified representation of a document . In this model, a text such as a sentence is represented as a set of words. The grammar is disregarded and even the order of words but multiplicity is kept same. In the modeling section, we will see how these concepts have been implemented to get the desired results using out model. We have considered only the last two educational degrees and last two work experiences for building the model.

3.5 Data Fields we used

Data fields taken into consideration for building the model:

- Name: Name of candidate
- Summary: As described in the About section of LinkedIn Profile
- Skills: Job profile related skills of candidate
- Experience_1: Experience of employee in the current organisation
- Current_Organisation: Latest organization of work
- Designation_1: Designation of employee in the last organization of work
- ExpDurationInMonths_1: Duration of Experience_1
- Designation_2: Designation of employee in the penultimate organization of work
- ExpDurationInMonths_2: Duration of Experience_2
- Education_1: Last degree of employee till date
- EduDurationInMonths_1: Duration of Education_1
- Education_2: Penultimate degree of employee till date
- EduDurationInMonths_2: Duration of Education_2
- Total_Exp: Total work experience
- Total_E: Total duration of education
- URL: LinkedIN Profile URL of candidate

Artificial Intelligence is the new electricity.

Andrew Ng

4
Data Modeling

MOST PEOPLE are familiar with patterns of word usage, how they are written, and how their arrangement dictate crossword puzzles and games of scrabble fitting together. Similarly, the pattern of usage of words is also closely linked when it comes to poems, prose, a news article or an advertisement. These intriguing patterns are not just the subject of lighthearted leisure; they also are the foundation of serious academic study. However, considering the way the modern world uses internet, it becomes helpful if we can segregate or cluster these “documents” based on some “topic”s so that searching becomes easier and relevant recommendations can be shown.

This project attempts to understand if the patterns studied in the semantics of the LinkedIn user profile combined with the linguistic insights of models can help cluster them under some topics or context and subsequently help to show candidate profiles as a search result to HR query based on relevance. This chapter introduces the field of topic modelling, which provides a formal approach for capturing document context.

4.1 Theory

Topic models[1] are hierarchical models of documents that explain an observed corpus with a small set of distributions over terms. When fitting to a corpus, these distributions tend to correspond to intuitive notions of the topics of the documents under consideration. Topic models are a powerful tool for unsupervised analysis of the text. The idea behind topic modelling is to imagine a probabilistic approach that attempts to guess a hidden thematic structure of the document. Given the observed collection, tries to determine the posterior distribution of the hidden thematic structure.

We can try to understand topic models as explained below three definitions.

- A topic model is a type of model for discovering the abstract topics that occur in a collection of documents.
- Topic models are a suite of algorithms that uncover the hidden thematic structure in the document collections. These algorithms help us develop new ways to search, browse and summarize large archives of text.
- Topic models provide a simple way to analyze large volumes of unlabeled text. A topic consists of a cluster of words that frequently occur together.

In a very simplistic manner a typical topic model would comprise of the below steps to arrive at a topic for a document:

- We only observe the documents
- Each document is a mixture of many topics

- Each topic is a distribution over words
- Each word is drawn from one of those topics

4.1.1 Latent Dirichlet Allocation

In natural language processing, the latent Dirichlet allocation (LDA) [2] is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. LDA is an example of a topic model. To infer the topics in a corpus, we imagine a generative process whereby the documents are created. Documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over all the words.

4.2 Our Workflow

4.2.1 Packages involved

The main package used to build the model is learnt. Spacy, NLTK packages are also used for text processing. Numpy and Pandas are used for storing and manipulating data in the python data frame.

4.2.2 Text Extraction

Each profile under consideration has been scraped using selenium and beautiful soup to get the relevant text information. Then using regex expressions, the field values have been extracted and loaded into a data frame for further analysis.

4.2.3 Data Cleaning

In this step, all types of punctuation were removed from the text data that was contained in the data frame.

4.2.4 Tokenization

Gensim's simple_preprocess() function was used to tokenize the data.Each sentence was converted to a list of words and unnecessary characters were removed.

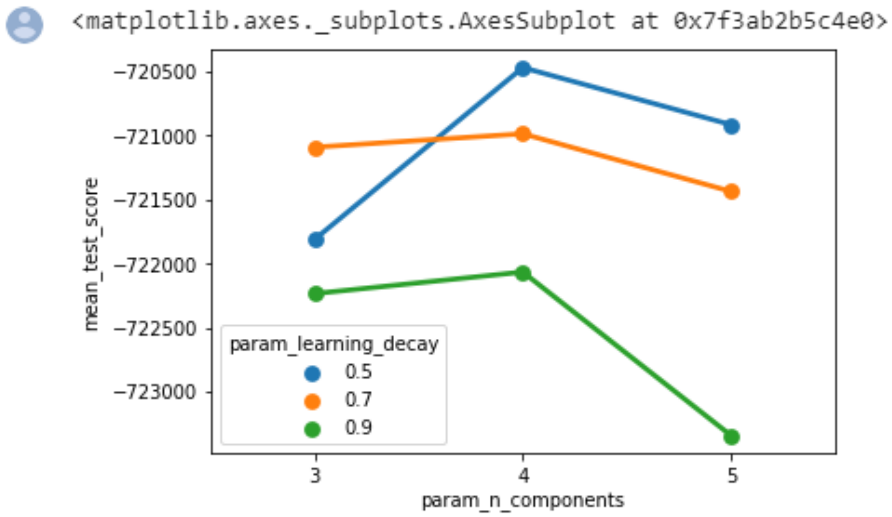
4.2.5 Lemmatization

This process finds the root form of the word being lemmatized. The advantage of this is, the total number of unique words in the dictionary reduces. Hence, the number of columns in the document-word matrix will be dense and with lesser columns.

4.2.6 Creation of document Word matrix

The LDA topic model algorithm requires a document word matrix as the main input. We can create one using CountVectorizer. In the below code, the CountVectorizer has been configured to consider words that have occurred at least 10 times (min_df), remove builtin English stopwords, convert all words to lowercase. So, to create the doc-word matrix, you need to first initialise the CountVectorizer class with the required configuration and then apply fit_transform to create the matrix. Since most cells contain zeros, the result will be in the form of a sparse matrix to save memory. If we want to materialize it in a 2D array format, call the 'todense' method of the sparse matrix like it's done in the next step.

Building LDA with sklearn. Now we can build a Latent Dirichlet Allocation (LDA), model. Let's initialise one and call fit_transform() to build the LDA model. We have set the number of topics as 7 based on prior knowledge about the data-set. Later we will find the optimal number using a grid search



```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3ab2b5c4e0>
```

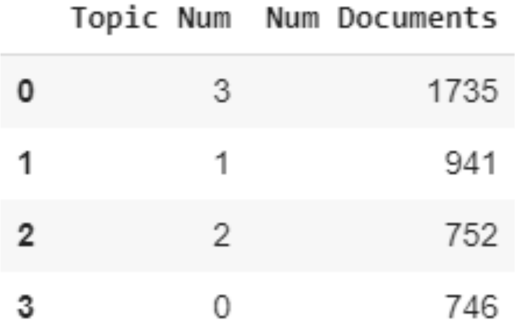
Figure 4.2.1: Grid Search Results

4.2.7 Finding the best model

Grid Search can be done to optimise the model to get its best version. The most important tuning parameter for LDA models is n_components (number of topics). Along with that, we would also search for learning_decay which controls the learning rate. The grid search will find multiple LDA models for all possible combinations of parameter values. We can then find the best model for our purpose and use it. The best model can be used by calling model. best_estimator_

4.2.8 Comparing LDA Scores

LDA Scores were compared and the best model was chosen. Refer Figure 4.2.1 which shows the comparison.



	Topic Num	Num Documents
0	3	1735
1	1	941
2	2	752
3	0	746

Figure 4.2.2: Topics against Documents

4.2.9 Finding dominant topic

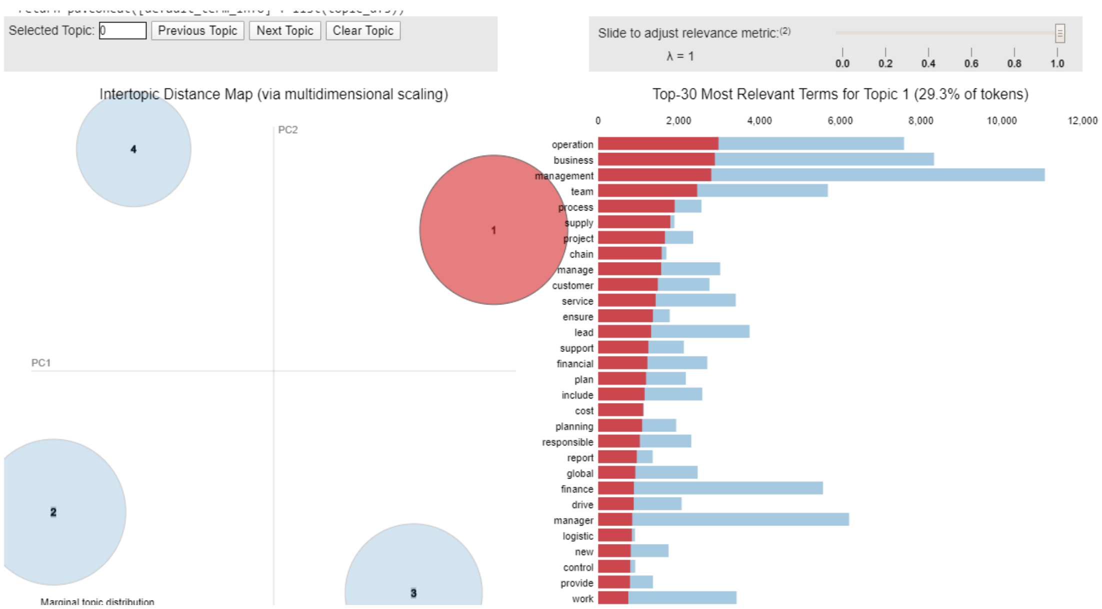
To classify a document as belonging to a particular topic, a logical approach is to see which topic has the highest contribution to that document and assign it. Refer Figure 4.2.2

4.2.10 Visualising LDA

Paul Davis offers the best visualization to view the topics-keywords distribution. A good topic model will have non-overlapping, fairly big sized blobs for each topic. This seems to be the case here. Refer Figure 4.2.3

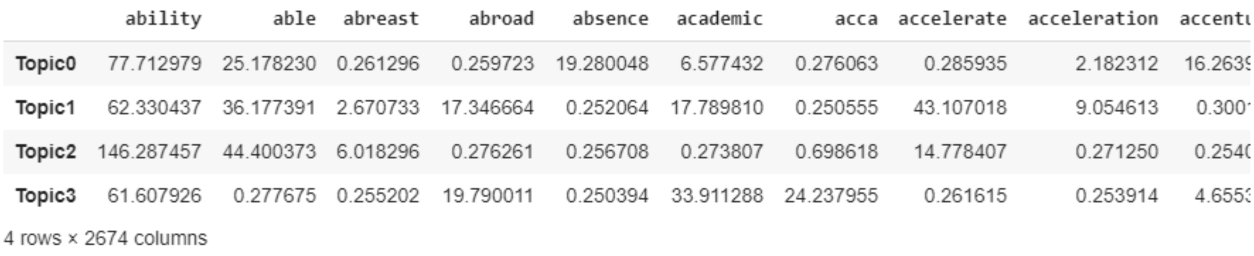
4.2.11 Obtained topics and its keywords

The weights of each keyword in each topic are contained in lda_model.components_as a 2d array. The names of the keywords itself can be obtained from the vectorizer object using functions. Refer Figure 4.2.4 and Figure 4.2.5 to check the topics.



```
return pd.concat([default_term_info] + list(topic_dfs))
```

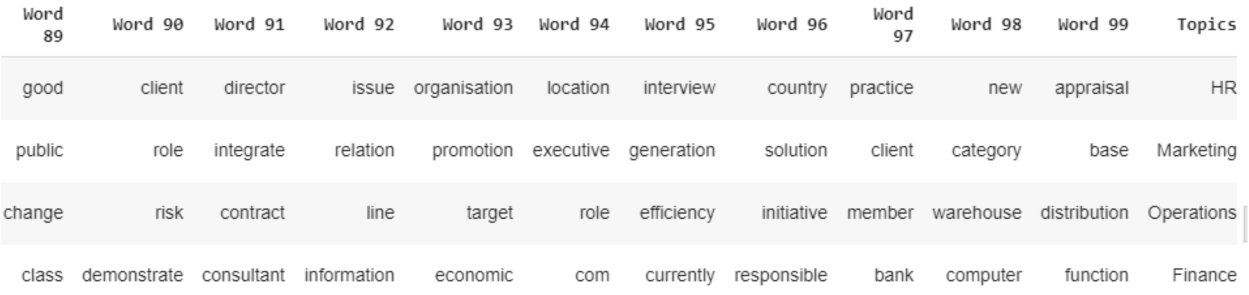
Figure 4.2.3: Topics LDA Visualisation



	ability	able	abreast	abroad	absence	academic	acca	accelerate	acceleration	accentu
Topic0	77.712979	25.178230	0.261296	0.259723	19.280048	6.577432	0.276063	0.285935	2.182312	16.2639
Topic1	62.330437	36.177391	2.670733	17.346664	0.252064	17.789810	0.250555	43.107018	9.054613	0.300
Topic2	146.287457	44.400373	6.018296	0.276261	0.256708	0.273807	0.698618	14.778407	0.271250	0.2540
Topic3	61.607926	0.277675	0.255202	19.790011	0.250394	33.911288	24.237955	0.261615	0.253914	4.6553

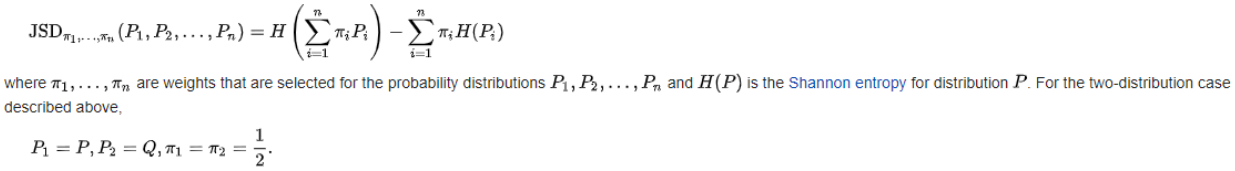
4 rows × 2674 columns

Figure 4.2.4: Topics Distribution



Word 89	Word 90	Word 91	Word 92	Word 93	Word 94	Word 95	Word 96	Word 97	Word 98	Word 99	Topics
good	client	director	issue	organisation	location	interview	country	practice	new	appraisal	HR
public	role	integrate	relation	promotion	executive	generation	solution	client	category	base	Marketing
change	risk	contract	line	target	role	efficiency	initiative	member	warehouse	distribution	Operations
class	demonstrate	consultant	information	economic	com	currently	responsible	bank	computer	function	Finance

Figure 4.2.5: Topics Selection



$$\mathrm{JSD}_{\pi_1,\ldots,\pi_n}(P_1, P_2, \ldots, P_n) = H\left(\sum_{i=1}^{n} \pi_i P_i\right) - \sum_{i=1}^{n} \pi_i H(P_i)$$

where $\pi_1, \ldots, \pi_n$ are weights that are selected for the probability distributions $P_1, P_2, \ldots, P_n$ and $H(P)$ is the Shannon entropy for distribution P. For the two-distribution case described above,

$$P_1 = P, P_2 = Q, \pi_1 = \pi_2 = \frac{1}{2}.$$

Figure 4.2.6: Jensen-Shanon Distance

4.2.12 Determining similarity between query and documents

Similarity[3] has been checked using Jensen-Shanon divergence(JSD) metric. In probability theory and statistics, the Jensen–Shannon divergence is a method of measuring the similarity between two probability distributions. The square root of the Jensen–Shannon divergence is a metric often referred to as Jensen-Shannon distance. Refer Figure 4.2.6

In god we trust; all others must bring data

William Edwards Deming

5

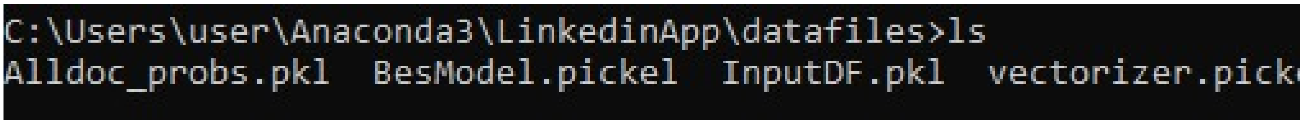
Model Deployment

Once the model has been built, it has to be deployed. The following steps are taken to deploy the model.

5.1 Our work flow

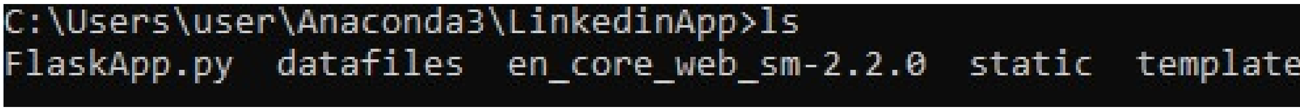
5.1.1 Creating the model file

The model file is saved as a pickle(.pkl) file to be used later. Pickling our model means to persist it by saving to disk so that we do not need to constantly retrain our models each time we want to use them.



```
C:\Users\user\Anaconda3\LinkedinApp\datafiles>ls
Alldoc_probs.pkl  BesModel.pickel  InputDF.pkl  vectorizer.picke
```

Figure 5.1.1: Pickle files



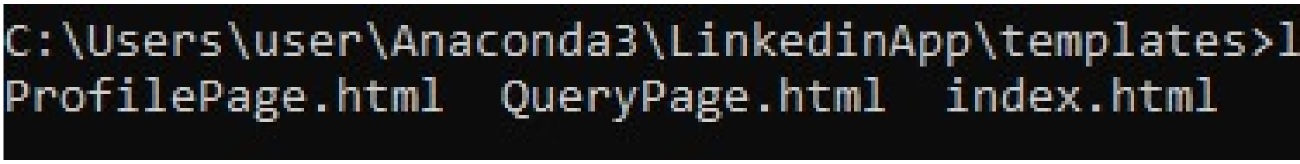
```
C:\Users\user\Anaconda3\LinkedinApp>ls
FlaskApp.py  datafiles  en_core_web_sm-2.2.0  static  template
```

Figure 5.1.2: Flask Components

5.1.2 Integrating Model with Flask

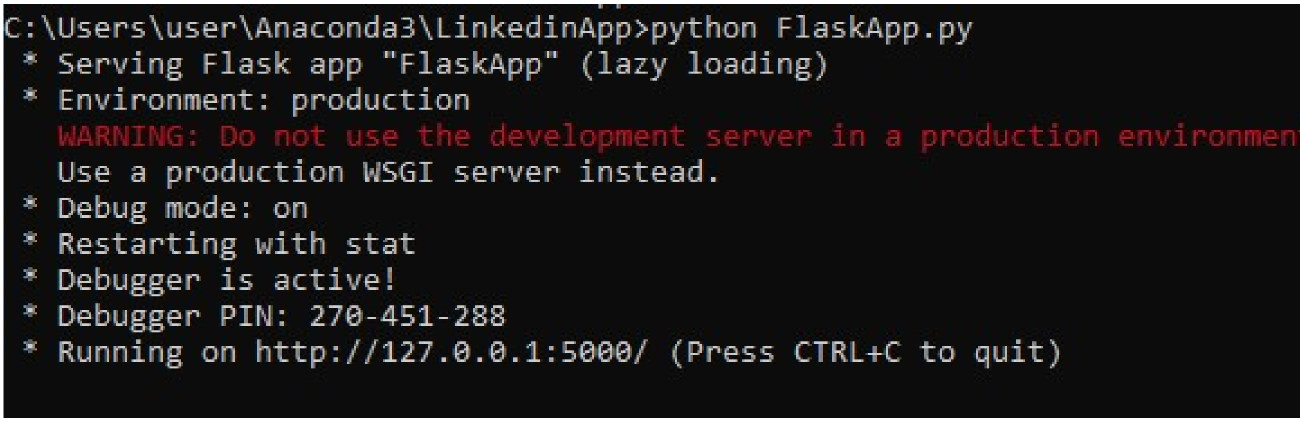
We need to create an API for a machine learning model, using Python along with Flask.This API will act as an access point for the model across many languages, allowing us to utilise the search capabilities through HTTP requests. The flask app consists of 2 main components:

- Python App
- HTML Templates



```
C:\Users\user\Anaconda3\LinkedinApp\templates>l
ProfilePage.html  QueryPage.html  index.html
```

Figure 5.1.3: HTML Templates



```
C:\Users\user\Anaconda3\LinkedinApp>python FlaskApp.py
 * Serving Flask app "FlaskApp" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environmen
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 270-451-288
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Figure 5.1.4: Flask integrated Python App

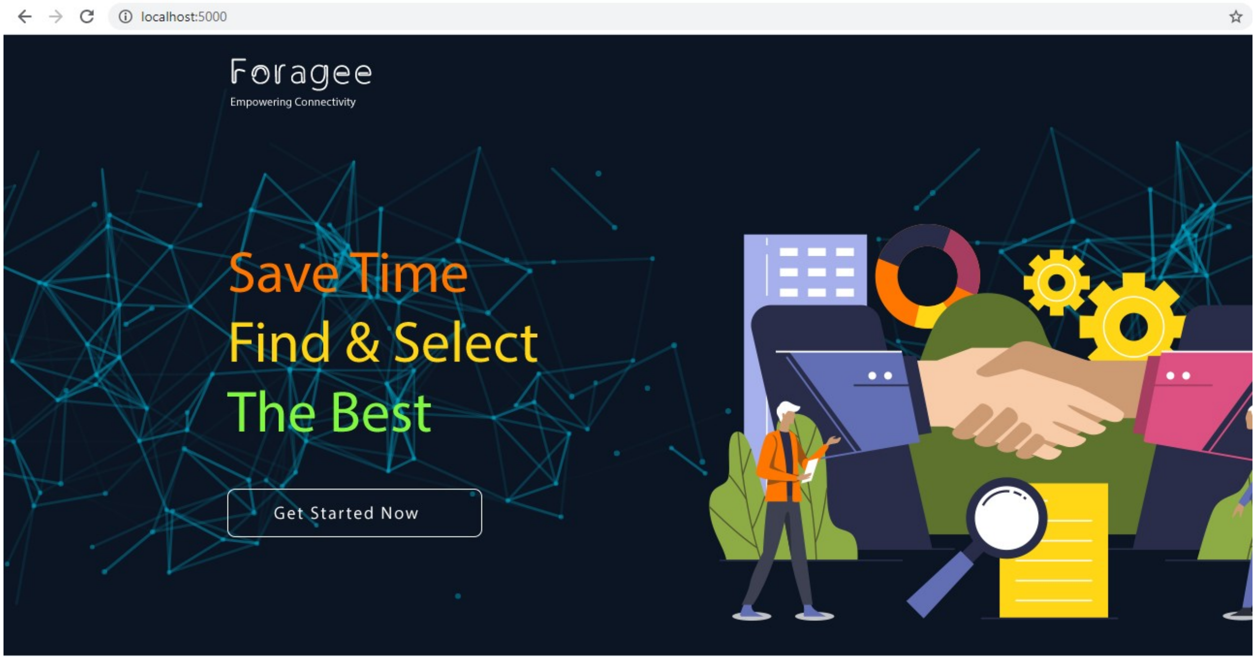
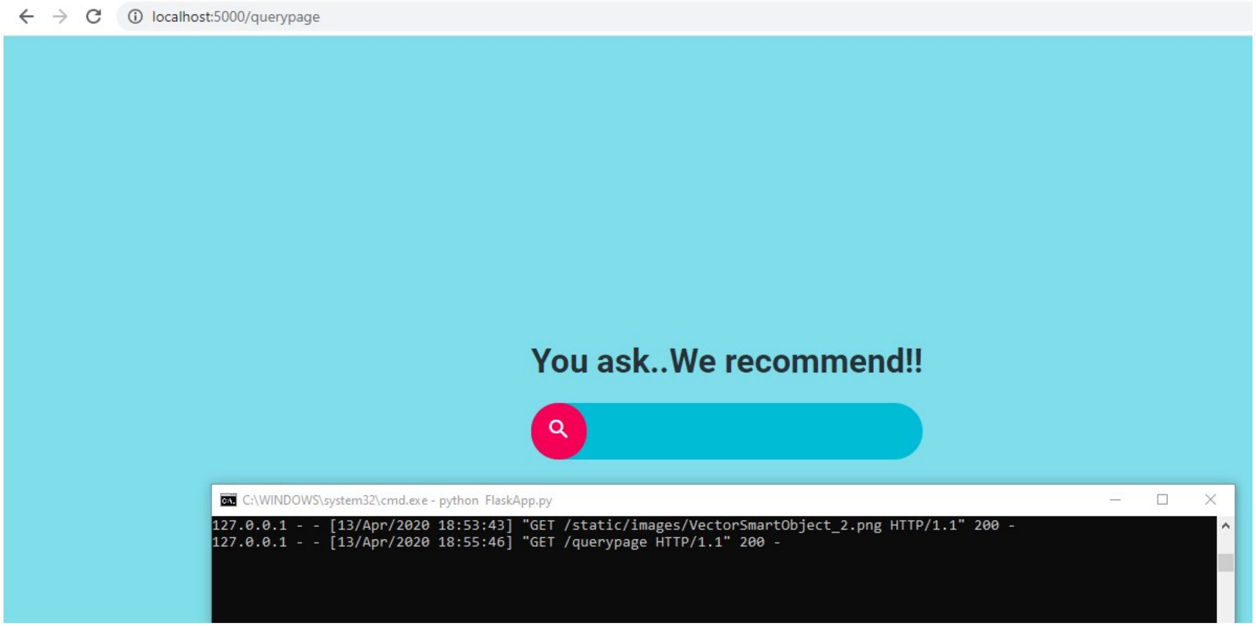


Figure 5.1.5: Forage Landing Page

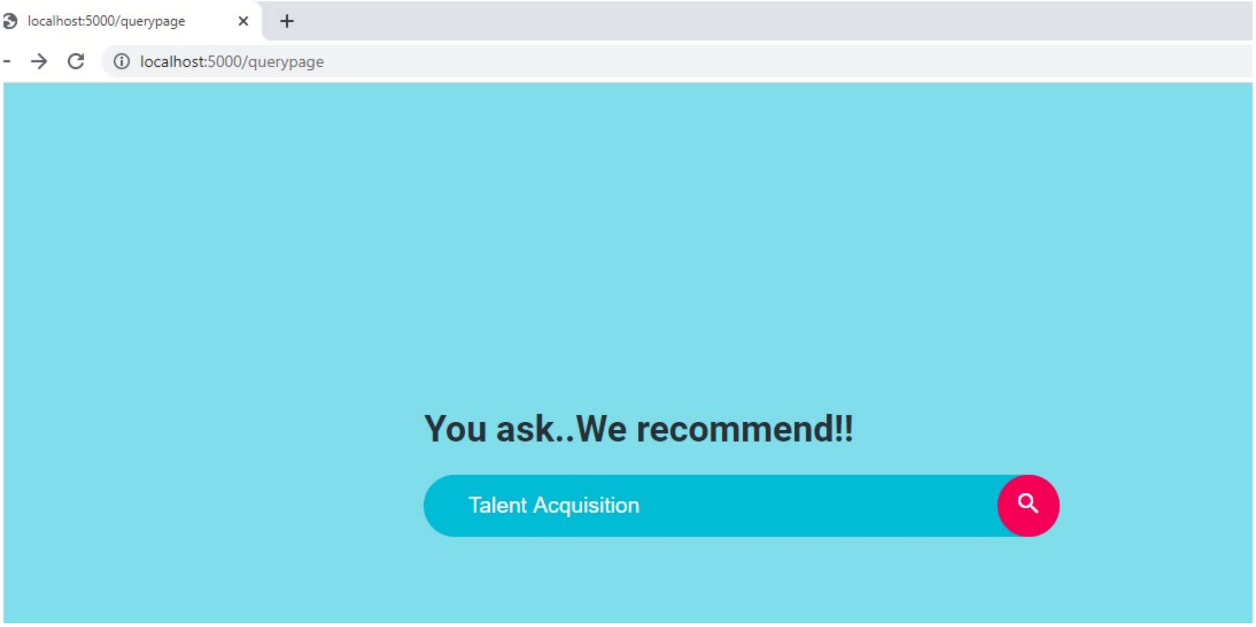


```
127.0.0.1 - - [13/Apr/2020 18:53:43] "GET /static/images/VectorSmartObject_2.png HTTP/1.1" 200 -
127.0.0.1 - - [13/Apr/2020 18:55:46] "GET /querypage HTTP/1.1" 200 -
```

Figure 5.1.6: Getting Started

5.1.3 Deployed on local machine

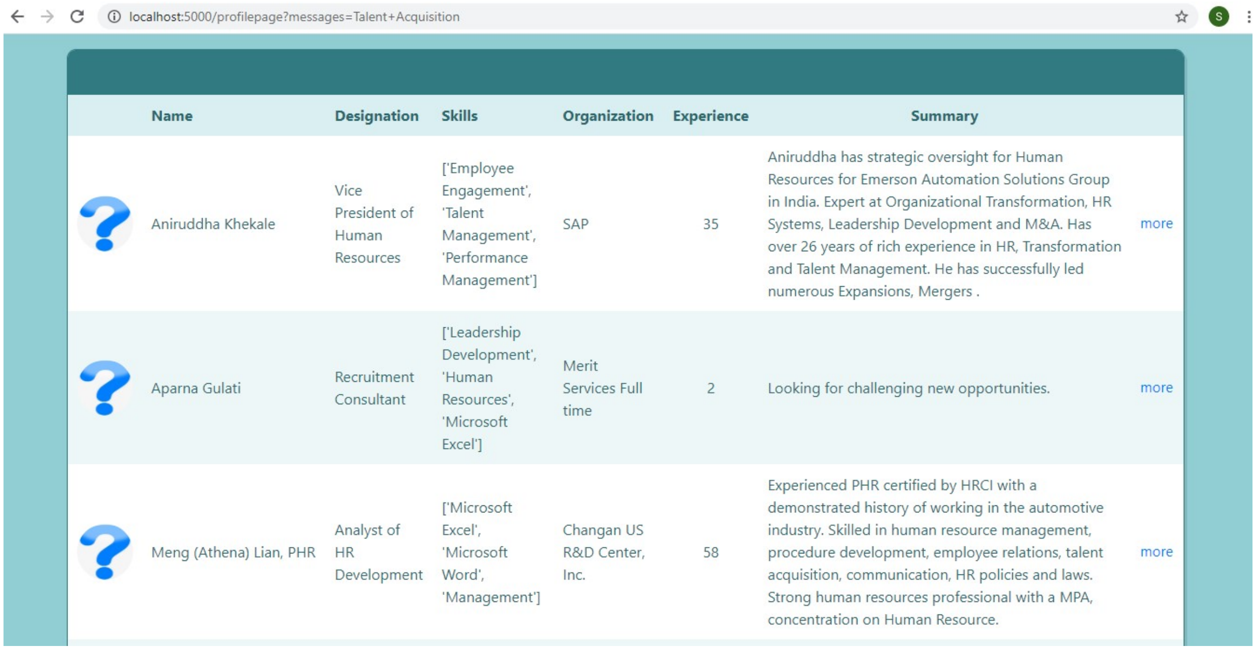
Application will be running on default port 5000



You ask..We recommend!!

Talent Acquisition

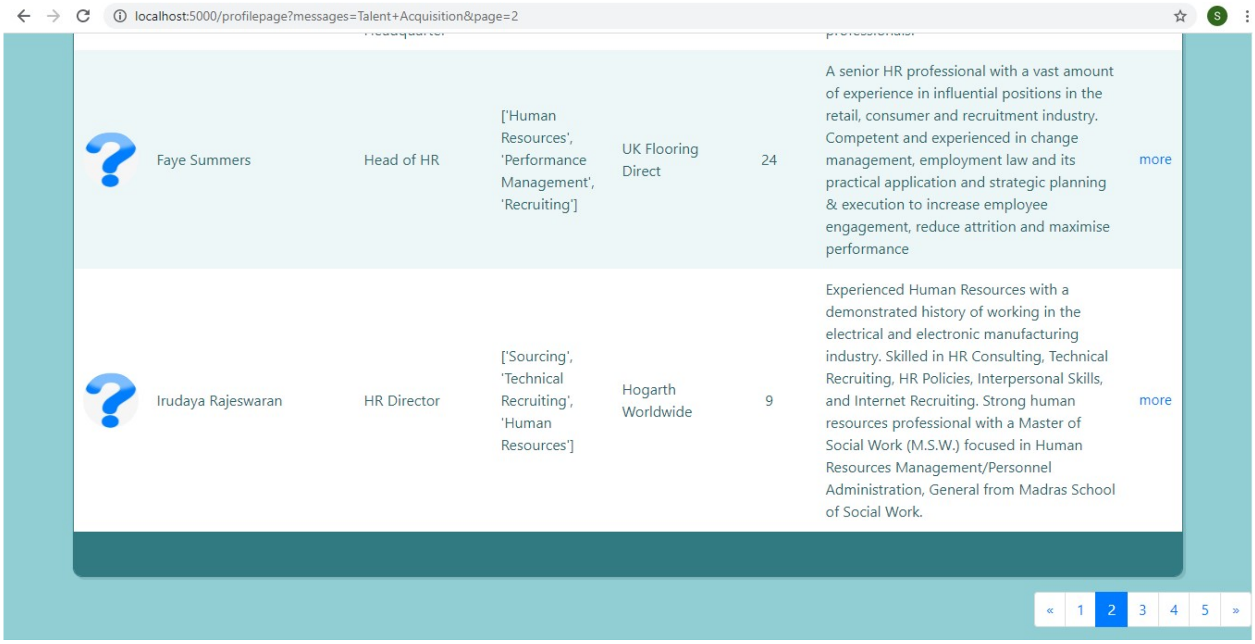
Figure 5.1.7: Search



localhost:5000/profilepage?messages=Talent+Acquisition

	Name	Designation	Skills	Organization	Experience	Summary	
❓	Aniruddha Khekale	Vice President of Human Resources	['Employee Engagement', 'Talent Management', 'Performance Management']	SAP	35	Aniruddha has strategic oversight for Human Resources for Emerson Automation Solutions Group in India. Expert at Organizational Transformation, HR Systems, Leadership Development and M&A. Has over 26 years of rich experience in HR, Transformation and Talent Management. He has successfully led numerous Expansions, Mergers .	more
❓	Aparna Gulati	Recruitment Consultant	['Leadership Development', 'Human Resources', 'Microsoft Excel']	Merit Services Full time	2	Looking for challenging new opportunities.	more
❓	Meng (Athena) Lian, PHR	Analyst of HR Development	['Microsoft Excel', 'Microsoft Word', 'Management']	Changan US R&D Center, Inc.	58	Experienced PHR certified by HRCI with a demonstrated history of working in the automotive industry. Skilled in human resource management, procedure development, employee relations, talent acquisition, communication, HR policies and laws. Strong human resources professional with a MPA, concentration on Human Resource.	more

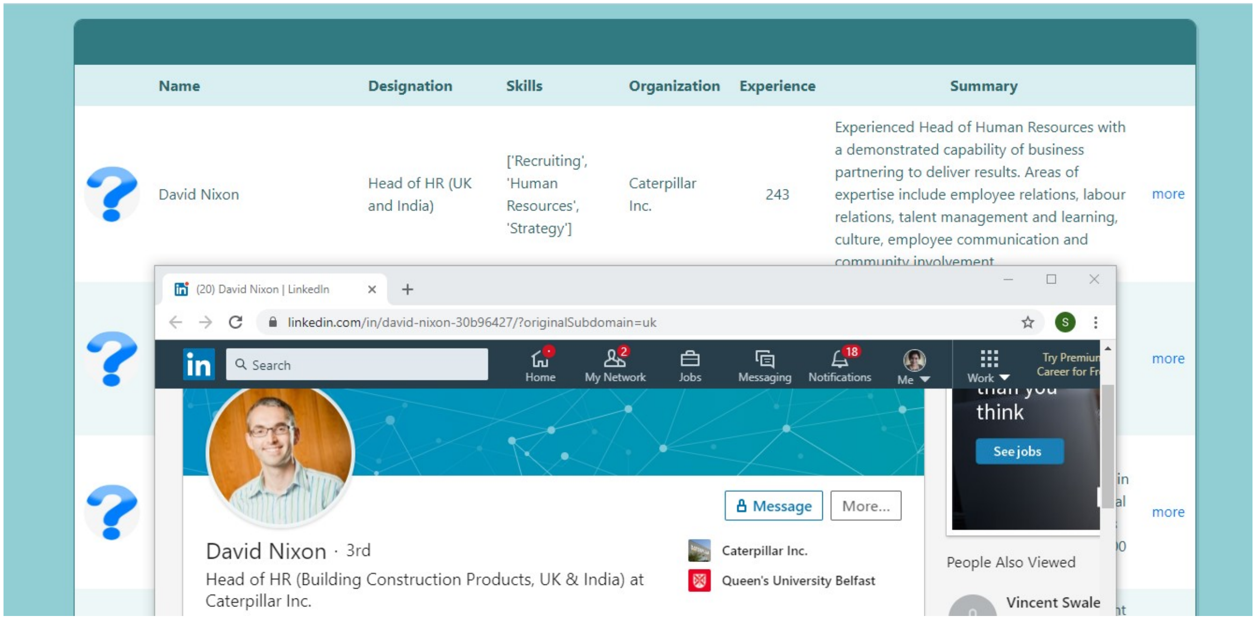
Figure 5.1.8: Search Results



localhost:5000/profilepage?messages=Talent+Acquisition&page=2

	Faye Summers	Head of HR	['Human Resources', 'Performance Management', 'Recruiting']	UK Flooring Direct	24	A senior HR professional with a vast amount of experience in influential positions in the retail, consumer and recruitment industry. Competent and experienced in change management, employment law and its practical application and strategic planning & execution to increase employee engagement, reduce attrition and maximise performance	more
	Irudaya Rajeswaran	HR Director	['Sourcing', 'Technical Recruiting', 'Human Resources']	Hogarth Worldwide	9	Experienced Human Resources with a demonstrated history of working in the electrical and electronic manufacturing industry. Skilled in HR Consulting, Technical Recruiting, HR Policies, Interpersonal Skills, and Internet Recruiting. Strong human resources professional with a Master of Social Work (M.S.W.) focused in Human Resources Management/Personnel Administration, General from Madras School of Social Work.	more

Figure 5.1.9: Pagination



	Name	Designation	Skills	Organization	Experience	Summary	
	David Nixon	Head of HR (UK and India)	['Recruiting', 'Human Resources', 'Strategy']	Caterpillar Inc.	243	Experienced Head of Human Resources with a demonstrated capability of business partnering to deliver results. Areas of expertise include employee relations, labour relations, talent management and learning, culture, employee communication and community involvement	more
							more
							more

(20) David Nixon | LinkedIn

linkedin.com/in/david-nixon-30b96427/?originalSubdomain=uk

David Nixon · 3rd

Head of HR (Building Construction Products, UK & India) at Caterpillar Inc.

Caterpillar Inc.

Queen's University Belfast

People Also Viewed

Vincent Swale

Figure 5.1.10: Actual Profiles

Data is the new oil

Clive Humbry

6 Conclusion

Work synopsis: The goal is to reduce the asymmetry of information between the recruiter pool and the candidates39; pool. We have used different NLP techniques and ML algorithms as our tool to acquire the most substantiated results. It took us almost three months to finally shape the model which we are currently looking at and it took immense scrutinizing and eye of details to inspect every bits and piece of the model starting from scraping till the deployment. Fortunately, with our decent promotional stances and pricing -few of them have turned up and sparks interest in our product- but to materialize it further, we need a considerable amount of time as we are still very naive (MVP).

In this project we tried to present a way in which the LinkedIn profiles can be searched in a such a manner that people with job roles similar to a job posting is presented to the HR personnel rather than only those profiles containing certain keywords same as used in the job description. To achieve this, we have used Topic Modelling and LDA to optimize the search based on a text query from the HR personnel. This particular methodology of searching profiles for a job will match the right resource with the right job roles more effectively and will thereby help in increasing business value for any organisation.

LIMITATIONS: Scraping is one of the bigger challenges as many anti scrapers cause stalling to the process. Due to time constrain we couldn't fetch much data as the number of data points that we used(5853) is much less than a corpus of 40K required for good topic modeling. The data which we collected are clustered into 4 generic segments - HR, Marketing, operations,and supply chain.The cluster could have been more granular had it been more data. For an advanced search, many more filters are added to enhance the accuracy of findings such as place, geography, experienced-based candidates for proper mapping of profiles. The data can get old by the time we fit into the model. As we are using a free subscription-based model, the standard protocol is to stop displaying after 1000 profiles or 100 pages but if we want to view all 10 k profiles in one go, we have to look for the relevancy of profiles only by selecting the advance filters.

SCOPE: We can train the model with more amount of data close to 40k to optimize the search results and offer better solutions. This particular application we can use for new emerging job profiles where the process is taking a paradigm shift. The model upon integrating with other job portals can help in finding the best one saving more time for an HR. The model can help in internal resource allocation in a consultancy firm, there are hundreds of people who juggle around different verticals who are being cross-trained and released now and then across various teams at different geographies for specific skill sets, once it integrates with the employees' portal of the organization, it can act as a centralized dashboard for all the managers looking for the right resource for project deliverable.

References

[1] HisaoTamakic SantoshVempalad Christos H.Papadimitrioua, PrabhakarRaghavanb. Latent semantic indexing: A probabilistic analysis. *Journal of Computer and System Sciences*, 61(2):217–232, 1999.

[2] Andrew Ng David Blei and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March,2003.

[3] Lillian Lee. Measures of distributional similarity. *Association of Computational Linguistics*, pages 99–1004.

[4] M.F. Porter. An algorithm for suffix stripping. *Mathematical Systems Theory*, 14(3):130–137, 1980.

Colophon

This thesis was typeset using LaTeX, originally developed by Leslie Lamport and based on Donald Knuth's TeX. The body text is set in 11 point Arno Pro, designed by Robert Slimbach in the style of book types from the Aldine Press in Venice, and issued by Adobe in 2007. A template, which can be used to format a PhD thesis with this look and feel, has been released under the permissive MIT (X11) license, and can be found online at github.com/suchow/ or from the author at suchow@post.harvard.edu.