Internship Boot Camp

CSD 350: Natural Language Processing Project

Prepared by Mehak Agrawal (1910110233)

ABSTRACT

In this paper, I have described the outline for my NLP project. This project helped me to implement my learning in class to actual real-time use. I have divided the project into 2 parts. The first half is based on the question bank corpus and the second part is on the resume corpus.

INTRODUCTION

While developing this project, I had 3 main goals in mind:

1. I wanted a project that would reinforce and apply the NLP concepts taught in the classroom.
2. I wanted to come up with a real-life solution for a problem all of us face.
3. I wanted to host the final product which could be used by students in future.

I have used NLP tools like Tokenization and stemming/lemmatization analysis of text corpus, stop word removal to modify infected words and analyze the entire document to build a structured dictionary, which one can use to save processing time.

LITERATURE REVIEW

1. **Research paper by Md Raihan Mia on Question Bank Similarity Searching System**
   https://www.researchgate.net/publication/331158767_Question_Bank_Similarity_Searching_System_QB3S_Using_NLP_and_Information_Retrieval_Technique

   This research paper is based on finding similar questions, handling duplicate questions and ranking search results of a query input based on NLP and Information Retrieval techniques. This project only helps to search questions of a particular topic without any help in the name of companies.

2. **Semantic similarity-based information retrieval as applied to MOOCs**
   https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1339&context=etd_projects

   The project uses a similarity phase approach and implements a vector space model to build a web page that takes a college's course details as input and compares items such as course title and description against MOOCs in order to identify similar courses.

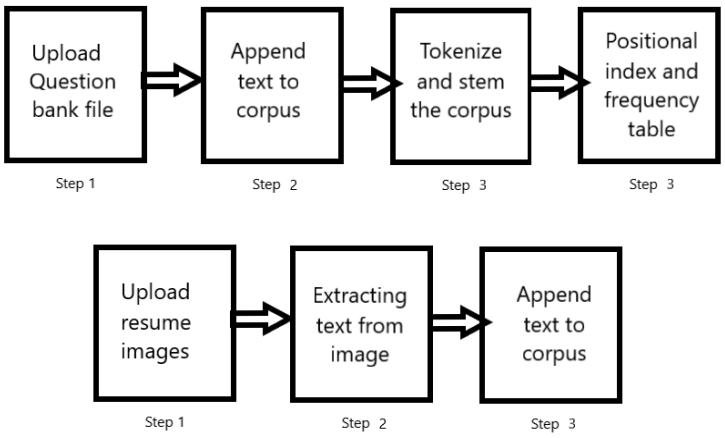
3. **A personalized resume-job matching system**
   https://core.ac.uk/download/pdf/79650795.pdf

   The project devises a new statistical-based ontology similarity measure to compare the resume model to the job models. They create a finite state transducer-based information extraction library to extract models from resume and job descriptions.

OBJECTIVE

Since the beginning of this year, all students in our batch have been trying and applying for internships. Soon, this will be changed to jobs. However, before studying and having technical knowledge for cracking interviews, the first step is to know what is asked in the interview and what should be present in your resume so that it gets selected for the coding/interview rounds of the company. These first steps are very important, yet students often skip them. Different companies focus on different topics according to the services and products they bring to the market, for example, Google focuses on 'Graph' while Dell focuses on 'SQL'. Therefore, while applying for internships at different companies, it is important for a student to know the topic he/she should focus on and include it in their resume.

PROPOSED MODEL



METHODOLOGY

1. **Top 10 topics asked in the company interviews**

   Query: Company Name
   Corpus: Question bank

   A frequency distribution table with the company name as col1 and frequency dictionary as col2 is produced. If the word in the dictionary exists in the technical topics asked in the interview, the frequency is saved in another dictionary – topics_dics. Using Counter, the top 5 most common topics asked by the company in a general interview are printed.

2. **Top 5 companies where the topic is asked maximum no of times**

   Query: Topic
   Corpus: Question bank

   A vector space model based on tf-idf which checks the cosine similarity of the topic for example stack with the doc, that is, company question bank file is implemented. Using Counter, the top 5 most common companies where the topic is often asked are printed.

3. **Check which company is the best fit for the candidate**

   Query: Resume
   Corpus: Question bank

   A vector space model based on tf-idf which checks the cosine similarity of the resume text extracted from the image with the doc, that is, the company question bank file is implemented. Using Counter, the top 5 most common companies where the candidate is the best fit are printed.

4. **Print top 10 candidates for the skill**

   Query: Skill
   Corpus: Resume

   A vector space model based on tf-idf which checks the cosine similarity of the company question bank file with the resume corpus is implemented. Using Counter, the top 5 best candidates for the company are printed.

Packages used:

1. nlltk – corpus, stemming, stop words
2. Pandas - frequency table
3. Collection - most_common
4. Pytesseract - extract text from image

Dataset used:

1. 47 Resume of current CSE Batch 2023
2. Question banks of 10 companies: 'Adobe', 'Amazon', 'Facebook', 'Flipkart', 'Google','GoldmanSachs, 'Microsoft', 'MorganStanley', 'Samsung','TCS'

Retrieval in the vector space model

- Query **q** is represented in the same way as a document.
- The term w_{iq} of each term t_i in **q** can also computed in the same way as in a document.
- Relevance of $\mathbf{d}_j$ to **q**: Compare the similarity of query **q** and document $\mathbf{d}_j$.
- For this, use cosine similarity (the cosine of the angle between the two vectors)
  - The bigger the cosine the smaller the angle and the higher the similarity

Dot product Unit vectors

$$\cos(\vec{q},\vec{d}) = \frac{\vec{q}\bullet\vec{d}}{|\vec{q}||\vec{d}|} = \frac{\vec{q}}{|\vec{q}|}\bullet\frac{\vec{d}}{|\vec{d}|} = \frac{\sum_{i=1}^{h} q_i d_i}{\sqrt{\sum_{i=1}^{h} q_i^2}\sqrt{\sum_{i=1}^{k} d_i^2}}$$

Where *h* is number of words (terms) in *q*, and *k* is the number of words (terms) in *d*.

EXPERIMENTATION AND RESULTS

Resume corpus

```
[41] ['diya', 'sachdev', 'ry', 'pree', 'logic', 'uae', 'rn', 'peer', 'etry', 'psat', 'education', 'shiv', 'nadar', 'university', 'majorin', 'computer', 'sci
     ['aakarsh', 'hajela', 'shiv', 'nadar', 'university', 'greater', 'noida', 'uttar', 'pradesh', 'computer', 'since', 'engineering', 'senior', 'secondary',
     ['pave', 'essa', 'eat', 'ee', 'aaa', 'reneur', 'skills', 'education', 'auwois', 'shiv', 'nadar', 'univesity', 'greater', 'noida', 'mojorin', 'computer'
     ['aarushi', 'dhir', 'ug', 'year', 'serisabod', 'haryana', 'google', 'drive', 'link', 'resources', 'ntasive', 'gosta', 'com', 'divers', 'vies', 'safina'
     ['aastha', 'dogra', 'rail', 'saurabh', 'colony', 'jabalpur', 'phone', 'email', 'linkedin', 'certifications', 'various', 'skills', 'available', 'kalai',
     ['aw', 'tree', 'visualizer', 'javafx', 'secret', 'auction', 'program', 'hotel', 'booking', 'system', 'using', 'php', 'ceaser', 'cipher', 'encoder', 'de
     ['akarsh', 'tyagi', 'education', 'shiv', 'nadar', 'university', 'major', 'computer', 'science', 'capa', 'delhi', 'public', 'schoo', 'ghaviabad', 'perce
     ['ccl']
     ['sse', 'aniket', 'gupta', 'q', 'bahadurgarh', 'university', 'roll', 'elprmeeceecoreoeemat', 'im', 'https', 'mike', 'gupta', 'projects', 'competitions'
     ['personal', 'info', 'address', 'ward', 'sector', 'phone', 'snu', 'id', 'email', 'linkedin', 'tps', 'linkedin', 'github', 'intps', 'skills', 'programm
     ['shraddha', 'arora', 'ff', 'education', 'shiv', 'nedar', 'university', 'unique', 'bachelor', 'technology', 'computer', 'science', 'engineering', 'cgp
     ['see', 'avantika', 'ln', 'ae', 'ey', 'eet', 'ore', 'icus', 'educational', 'history', 'shiv', 'nadar', 'university', 'nehru', 'world', 'school', 'brigh
     ['ayush', 'jagga', 'ca', 'seer', 'lecaat', 'fey', 'oe', 'tn', 'summary', 'skills', 'cam', 'ayer', 'amon', 'oka', 'mre', 'aout', 'tend', 'esting', 'more
     ['hb', 'avush', 'varma', 'mail', 'github', 'linkedin', 'education', 'skills', 'shivnadar', 'university', 'programming', 'bachelor', 'engineering', 'cor
     ['devyansh', 'sehgal', 'computer', 'science', 'undergrad', 'shiv', 'nadar', 'university', 'roll', 'tray', 'al', 'based', 'checkers', 'game', 'nee', 'ap
     ['jaisurya', 'rs', 'festnou', 'ed', 'sn', 'gthab', 'uaa', 'chena', 'ra', 'education', 'objective', 'rich', 'compuer', 'sass', 'eapowrag', 'cpt', 'sce',
     ['keshav', 'khandelwal', 'varanas', 'india', 'ttl', 'linkedin', 'projects', 'competitions', 'skills', 'ot', 'uses', 'automobile', 'hcl', 'tech', 'septe
     ['khushi', 'chawl', 'rgrad', 'shiv', 'nadar', 'university', 'opportunities', 'undergraduate', 'research', 'checkers', 'gui', 'game', 'using', 'al', 'ga
     ['khushi', 'nayal', 'ameticulous', 'concise', 'dedicated', 'person', 'always', 'strives', 'learn', 'something', 'whatever', 'always', 'looking', 'towar
     ['ns', 'vijayawada', 'ap', 'krushwant', 'koppolu', 'rushwant', 'projects', 'skills', 'java', 'food', 'ordering', 'shiv', 'nadar', 'university', 'built'
     ['reach', 'mobile', 'oftcial', 'ema', 'snueduin', 'linkedin', 'linkedin', 'koppuravur', 'address', 'maruthi', 'plaza', 'ra', 'new', 'marathi', 'nagar',
     ['coster', 'noi', 'ia', 'ln', 'inked', 'cominav', 'mare', 'othub', 'skills', 'aac', 'oe', 'ac', 'oi', 'si', 'tinea', 'agere', 'oa', 'oo', 'languages',
     ['ush', 'goyal', 'projects', 'event', 'calendar', 'javafx', 'mysql', 'april', 'created', 'desktop', 'calendar', 'application', 'importing', 'calendarfx
     ['q', 'gurueram', 'haryana', 'projects', 'skills', 'core', 'java', 'event', 'march', 'cch', 'created', 'desktop', 'calendar', 'application', 'using',
```

Frequency table

```
[22] pd.DataFrame(list(freq.items()), columns = ["Doc","Frequency"])
```

	Doc	Frequency
0	Google	{'general': 11, 'dfs': 6, 'oops': 2, 'web': 4,...
1	Microsoft	{'string': 7, 'array': 20, 'linkedlist': 7, 't...
2	Amazon	{'coding': 1, 'write': 11, 'efficient': 5, 'pr...
3	Facebook	{'general': 3, 'hashing': 3, 'string': 11, 'bi...
4	Adobe	{'write': 9, 'efficient': 1, 'program': 2, 'co...
5	Flipkart	{'given': 36, 'array': 30, 'integer': 6, 'b': ...
6	MorganStanley	{'given': 31, 'array': 33, 'consisting': 1, 't...
7	TCS	{'exam': 4, 'conducted': 1, 'following': 1, 'a...

Query - Company Name
Print top 5 most asked topics in their interviews

```
['Adobe', 'Amazon', 'Facebook', 'Flipkart', 'Google', 'Microsoft', 'MorganStanley', 'TCS']
Enter company name Flipkart

Flipkart

array
string
stack
linkedlist
matrix
```

Query - Topic
Print top 5 companies which ask the particular topic the most

```
['General', 'DFS', 'OOps', 'Web', 'DBMS', 'String', 'Error', 'Java', 'LinkedList', 'BFS', 'sorting', 'stack', 'matrix', 'python', 'C', 'Array', 'OS', '
LinkedList
workss
['linkedlist']
{'Adobe': 1.9631129329099652, 'Amazon': 2.0888951238169855, 'Facebook': 1.8503836364759405, 'Flipkart': 2.045322978786657, 'Google': 2.087375724662896,
Amazon  2.0888951238169855
Google  2.087375724662896
Flipkart  2.045322978786657
MorganStanley  1.992490398221254
Adobe  1.9631129329099652
```

Query - Resume 46
Print top 5 companies best fit for the candidate

```
[37]
['vamshidhar', 'reddi', 'pulugu\n\ncontests)', 'b\n\nproject', '/', 'competitions\n\n', '\n\n@', 'librari', 'manag', 'system\n+', 'thisprojecti', 'desi
{'Adobe': 10.762976187664183, 'Amazon': 8.51145096603597, 'Facebook': 6.816654537182469, 'Flipkart': 8.80393987029132, 'Google': 17.805166166822474, '
```

```
[38] from collections import Counter

     d = Counter(result)
     d.most_common()

     flag = 0
     for k, v in d.most_common(5):
       if v!= 0:
         print (str(k) + "  " + str(v))
```

```
Google  17.805166166822474
Adobe  10.762976187664183
MorganStanley  10.270374652816892
Flipkart  8.80393987029132
Amazon  8.51145096603597
```

5s completed at 3:52 PM

Query - machine learning
Print top 10 resumes with maximum cosine similarity with the skill query

```
machine learning
['machin', 'learn']
{0: 1.8390548085213219, 1: 1.6896995288781609, 2: 1.1722136039924793, 3: 0.0, 4: 1.855755875770409, 5: 1.461043842372023, 6: 1.9563574276609343,
Resume No 39
Resume No 33
Resume No 7
Resume No 10
Resume No 5
Resume No 1
Resume No 13
Resume No 14
Resume No 45
Resume No 9
```

CONCLUSION AND LIMITATIONS

In this project, I set out to design and construct a front-end system that would enable weighted searching with relevant feedback to be carried out on a conventional vector space model.

The system certainly has limitations and requires development in certain directions. Firstly, the front-end hostable website is yet to be developed. Secondly, one needs to manually upload the corpus to the database for use. Thirdly, the question bank corpus should be developed in a particular order to get good results.

Nevertheless, the system with a little further development is hostable onto the internet and can be used to help students with their interview process.

REFERENCES

1. https://www.researchgate.net/publication/331158767_Question_Bank_Similarity_Searching_System_QB3S_Using_NLP_and_Information_Retrieval_Technique
2. https://core.ac.uk/download/pdf/79650795.pdf
3. https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1339&context=etd_projects
4. https://www.machinelearningplus.com/nlp/cosine-similarity/
5. https://drive.google.com/drive/u/1/folders/1RsLFYzJZrRy4_yNCIbUVvjh_RguhWtBn
6. https://www.geeksforgeeks.org/