Resume and Job Description Matching

UMD MSML641 Final Project

By Alem Getu,
Loza Mengistu,
Zernab Gohar

Abstract

In this project, we explore the use of natural language processing (NLP) techniques to automate the process of job description to resume matching. We utilize the popular NLP technique Doc2Vec to extract relevant features from the job description and resume documents, and then use a similarity measure to find the best matches for resume and job description pairings. Our experiments on a real-world dataset of job descriptions demonstrate promising results for our final model's performance compared to our implementations of simpler baseline solutions. While we were unable to conclusively prove its superiority due to the absence of ground truth labels for comprehensive evaluation, our findings suggest potential practical applications in talent acquisition at companies who wish to streamline their hiring processes.

Introduction and Background

The task of matching resumes to job descriptions is a time and resource consuming process, both for employers and job seekers alike. Employers may receive hundreds of resumes for each job opening, making a manual screening process incredibly challenging, (if not impossible). Similarly, job seekers also must sift through an overwhelming number of job postings, often across multiple platforms. Job seekers receive limited feedback on why their applications are rejected and how they may tailor their resumes more effectively to increase their chances of a job match. NLP techniques may help reduce these problems by extracting key information from resumes and job descriptions to automate and improve the quality of matches.

Datasets

For our data sources, we utilized two existing datasets to train and test the model in our experimentation. For the job description dataset, we came across the “Indeed (USA) Job Listings Dataset”, a dataset consisting of over 30k records created from real job listings on Indeed.com that was collected using web scraping and data mining techniques.

Datasets available for resume data, however, were smaller in size and harder to find. It was especially difficult to find resume datasets with a variety of different job industries, as many would be a collection of specialized jobs that did not map well to the more broader job categories of the Indeed job description dataset. This caused our group difficulties down the line and led us to gain valuable insight into the significance of clean and well-balanced data.

Throughout the course of this project, we came to truly appreciate the often repeated motto that "a model is only as good as the data it is fed".

We tried out a couple of resume datasets, but the best resume dataset for our problem proved to be a publicly provided dataset collected by SurgeAI called "Public Resume Categorization". It contains 575 records, and while still smaller than the Indeed job description dataset and some of the other resume datasets we came across, it was the most class-balanced resume dataset we found that had a variety of job category classes.

Datasets used included:

- *Indeed (USA) Job Listings Dataset* [https://data.world/promptcloud/indeed-usa-job-listing-data]
    - Extension: ldjson
    - Size: 30k records (job listings)
    - Available Fields : uniq_id, crawl_timestamp, url, job_title, ***category***, company_name, logo_url, city, state, country, post_date, ***job_description***, job_type, apply_url, company_description, job_board, geo, job_post_lang, inferred_iso2_lang_code, extra_fields, is_remote, test1_cities, test1_states, test1_countries, site_name, html_job_description, domain, postdate_yyyymmdd, predicted_language, inferred_iso3_lang_code, test1_inferred_city, test1_inferred_state, test1_inferred_country, inferred_city, inferred_state, inferred_country, has_expired, last_expiry_check_date, latest_expiry_check_date, dataset, postdate_in_indexname_format, segment_name, duplicate_status, job_desc_char_count, ijp_reprocessed_flag_1, ijp_reprocessed_flag_2, ijp_reprocessed_flag_3, fitness_score

  For our project we extracted the two fields ***category*** and ***job_description.***

- *Resume Corpus Dataset by Github user florex* [https://github.com/florex/resume_corpus]
    - Extension: txt
    - Size: 19K records
    - Available Fields: reference_id, list_of_occupations, **resume_text**

- *Hugging Face Resume Dataset by user Sachinkelenjaguri* [https://huggingface.co/datasets/Sachinkelenjaguri/Resume_dataset]
    - Extension: imported using from datasets import load_dataset
    - Size: 962
    - Available Fields: **Category**(string), **Resume**(String)

- *Resumes and Job Categorization Dataset* [https://www.surgehq.ai/datasets/resumes-and-job-categorization-dataset]
    - Extension: .csv (json was also available for download)
    - Size: 575 records

- Available Fields: Res, task_id, task_response_id, ID, **Resume_str**, Resume_html, What is their work experience level?, What best describes their education level?, **What industry and sub-sector (most recent work) best describes this resume?**, You selected "N/A: Other Industry / Other Category".....write in the best description (1-3 words) of what industry should be?

Based on the labels for this particular dataset, it seems it was collected via a survey.

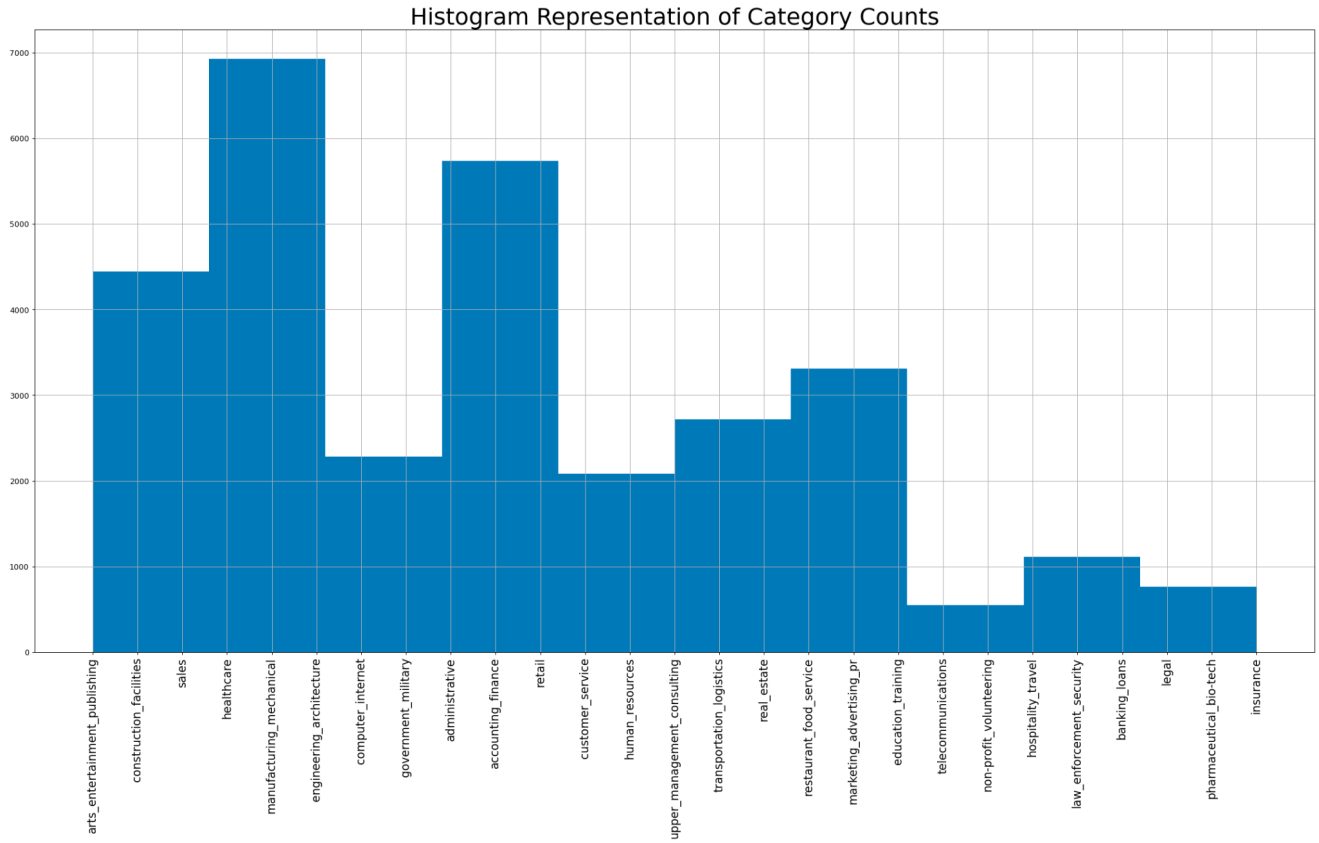
In addition, we downloaded 5 sample pdfs of resumes from the internet to perform pdf text extraction on.

For data cleaning and preprocessing, we performed the following:

- Removed punctuations
- Striped multiple whitespaces
- Removed non alphanumeric characters
- Converted all to lowercase

Understanding Our Dataset

Our job description dataset has 27 categories (classes):
{'arts_entertainment_publishing': 274, 'construction_facilities': 2227, 'sales': 1946, 'healthcare': 3647, 'manufacturing_mechanical': 2250, 'engineering_architecture': 1029, 'computer_internet': 2183, 'government_military': 99, 'administrative': 3576, 'accounting_finance': 1036, 'retail': 1125, 'customer_service': 1414, 'human_resources': 672, 'upper_management_consulting': 680, 'transportation_logistics': 1853, 'real_estate': 183, 'restaurant_food_service': 1722, 'marketing_advertising_pr': 637, 'education_training': 949, 'telecommunications': 182, 'non-profit_volunteering': 364, 'hospitality_travel': 327, 'law_enforcement_security': 318, 'banking_loans': 470, 'legal': 228, 'pharmaceutical_bio-tech': 204, 'insurance': 333}

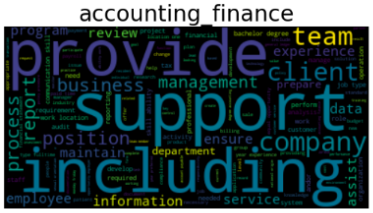


We notice from the above histogram that our class distribution in the dataset is **NOT balanced.** For example, the healthcare category has 3647 records but telecommunication has only 182 records, that is about a 20:1 ratio.

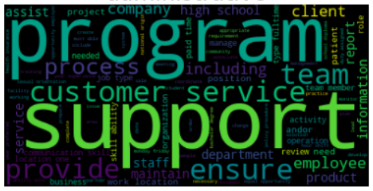
We then used wordCloud to view the most frequent words in each of the categories (classes).

From our analysis, we noticed that there are some words that do not add value to their respective category. We later added these words to our stop words list.

accounting_finance



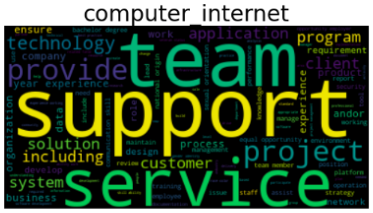
administrative



arts_entertainment_publishing

banking_loans

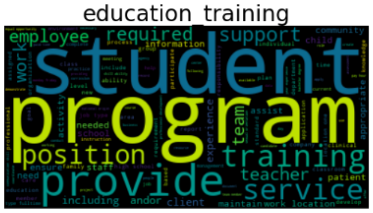
computer_internet



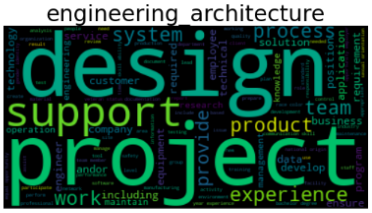
construction_facilities

customer_service

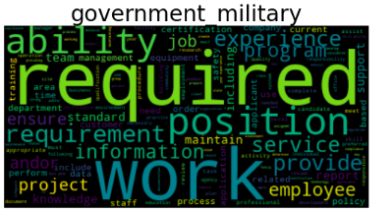
education_training



engineering_architecture



government_military



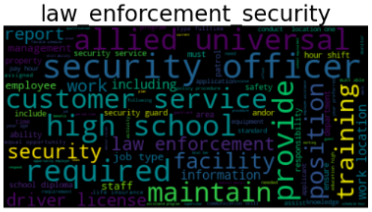
healthcare

hospitality_travel

human_resources

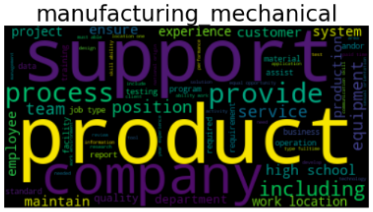
insurance

law_enforcement_security



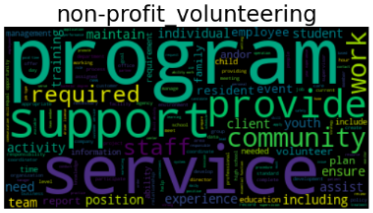
legal

manufacturing_mechanical

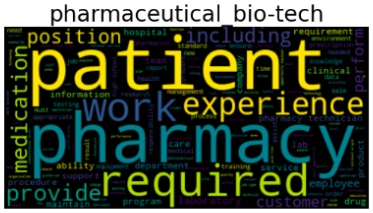


marketing_advertising_pr

non-profit_volunteering



pharmaceutical_bio-tech



real_estate

restaurant_food_service

retail

sales

telecommunications

transportation_logistics

upper_management_consulting

The Baseline System(s) and Their Implementation

Naive Word Matching

The initial baseline approach we implemented was Naive Word Matching, where we compared resumes with job descriptions based on the similarity of their word contents. This technique does not incorporate any contextual or semantic understanding of language. It solely relies on word frequencies to find matches, and is therefore a rudimentary approach to this problem. However, we wished to include it in our experimentation as a starting point to compare our final Doc2Vec model to and to gain a better understanding of what to improve upon in our evaluation metrics.

To ensure we were producing meaningful comparisons of text, we first preprocessed the data by applying removal of punctuation, lowercasing, etc. We perfected preprocessing techniques we applied over the span of the project, eventually adding our own stopwords file of words that we noted were frequently used throughout all job description documents, regardless of job category.

We used CountVectorizer from the scikit-learn library, as it counts the frequency of each word in the text and automatically tokenizes the input text. CountVectorizer converts the textual data into a matrix representation, where each row corresponds to a single resume or job description document, and each column represents a unique word or term present in the corpus.

We then applied cosine similarity using sklearn.metrics.pairwise to measure the similarity between the resume and job description vectors.

The matched pairs were then saved into a text file in the following format:

Resume id: 7
Resume Category: Retail / Manager or Store Keeper (includes clerk, cashier, or sales staff)

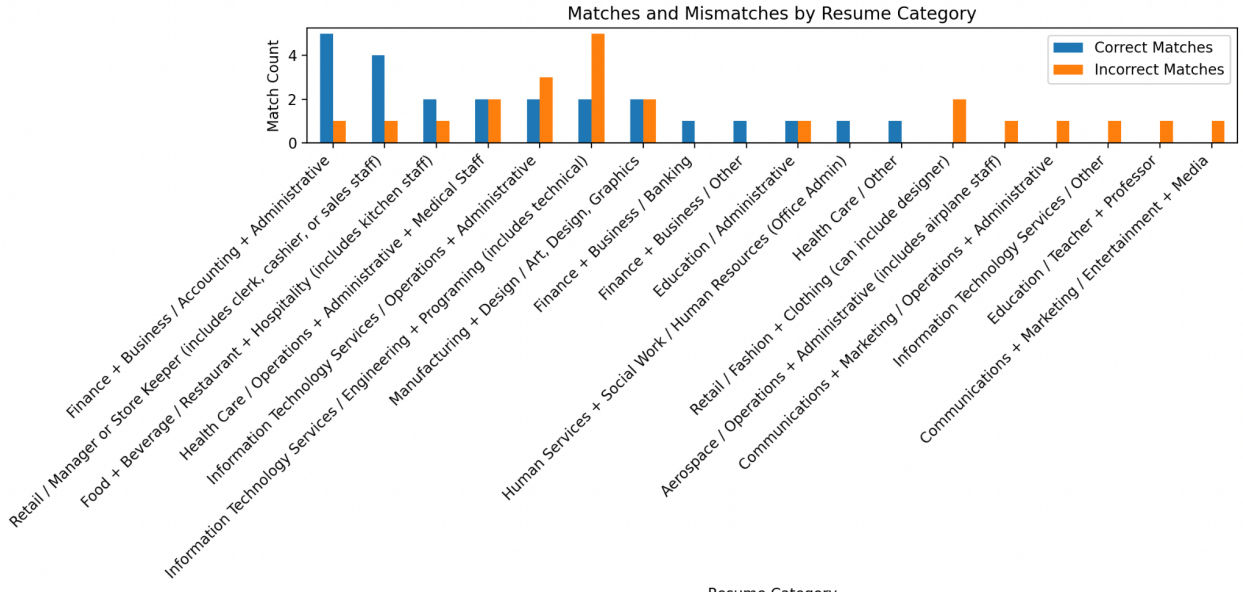
Job Desc id: 594
Job Description Category: sales

(Similarity Score: 0.7625797257826146)

Upon examining the text file of the matched pairs, we could only make approximate manual evaluations on how well Naive Word Match was doing. To gain a better understanding of how well this technique was performing matches, we needed some kind of ground truth label shared amongst both the resume and job description datasets that would truly give us insight on its performance.

We resorted to a manual annotation approach where we created a subset from matched pairs (consisting of 50 pairs: 25 positive matches, 25 non-matches), and added a new column “Label” to act as our ground truth label. A “0” represented a mismatch, while a “1” represented a

match for the record pair. While the annotation process provided slight insight into the performance of the technique, the process of manual annotation proved to be very tricky. The subset of data being only 50 matches was not a good representation of the entire dataset. Also, the way we selected which matched pairs would go into our subset was random manual selection, and that must obviously have skewed the class counts for Matches/Mismatches. As a result, the obtained results should be interpreted with caution:



The obtained results do not provide a conclusive assessment of Naive Word Matching's effectiveness compared to alternative approaches. While the manual annotation results may have fallen short, this experience enhanced our understanding of annotation processes and the efforts that go into them. It really motivated us to explore alternate strategies to overcome the challenges of manual annotation.

TF-IDF

Next, we look towards TF-IDF to enhance our matchings. While Naive Word Matching solely looks at word frequencies in each individual document, TF-IDF takes into account both the local relevance of a word within a document, (term frequency), and the global importance of the word across all documents, (inverse document frequency). TF-IDF also does a much better job at handling common words, as it reduces the weightage of more commonly used words in a document due to inverse document frequency, allowing the model to focus on the more meaningful terms.

As with the Naive Word Matching approach, we stayed consistent with our data preprocessing steps. Instead of CountVectorizer, we used TfidfVectorizer to convert the preprocessed resumes and job descriptions into their numerical representations by calculating the TF-IDF weight for each word in the documents.

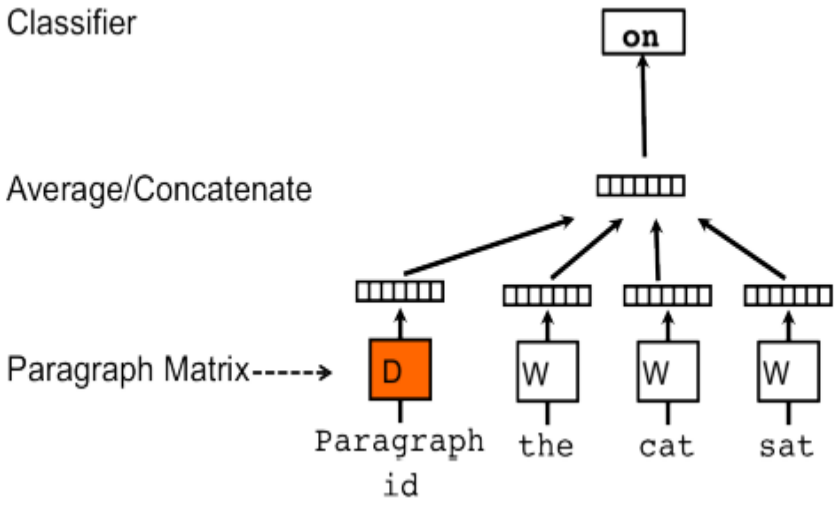
By incorporating term frequency and inverse document frequency, TF-IDF allowed us to prioritize the more relevant and meaningful terms while reducing the influence of common words. However, despite this, we encountered the same challenges in mapping resumes to job descriptions using TF-IDF that we did in Naive Word Matching.

Doc2Vec

As with our previously implemented techniques, we utilized the power of vectorization to compute cosine similarity. However, for this particular approach, we chose a different vectorization method known as Doc2Vec, sourced from the gensim library.

Doc2Vec builds upon the foundation of Word2Vec. It transforms our job descriptions and resumes into numeric vector representations as with the previous vectorization techniques we used, except Doc2Vec's vector representations allow us to measure the *semantic similarity* between resumes and job descriptions. Doc2Vec also considers the context of words and phrases when performing vectorization. In contrast, TF-IDF treats each word in isolation, without considering the context in which it appears.

This is an improvement from our previous approaches, as neither Naive Word Matching nor TF-IDF can capture semantic similarity. Using Doc2Vec's vectorization, it is also possible to use clustering algorithms to group similar documents together with the idea that documents with similar content will have vectors that are fairly close to each other.



Our Doc2Vec Model

After we cleaned and tokenize our job descriptions datasets we used it to train our Doc2Vec Model using: (*code: main_resume_jd_matching.ipynb*)

- Doc2Vec algorithm from the gensim library
- Using PV-DM technique (Distributed Memory version of Paragraph Vector)
- 70/30 train/test split

- vector_size=50
- epochs=50
- min_count=10

The main goal of our model is to find a job posting that matches most to a resume or vice versa. So we used our model to infer vectors for some of the sample resumes in our resume datasets and calculated the cosine similarity and evaluated the results by eye.

For the most part for the resumes and job description we manually looked at the cosine similar score we got from our model made sense. For example, the resume of a data scientist and a job posting for a table waiting position (in restaurant and food category) we get a cosine score of 0.004.

However, we wanted a better way to see if our model is working as intended.

Assessing our Model

To assess our new 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. We are basically pretending as if the training corpus is some new unseen data and seeing how they compare with our model with the expectation of overfitting.

Out of 20948 inferred documents 20492 were found to be most similar to another document. That means greater more than 97% of the documents were matched to themselves and the remaining 3% were matched as most similar to other documents by mistake

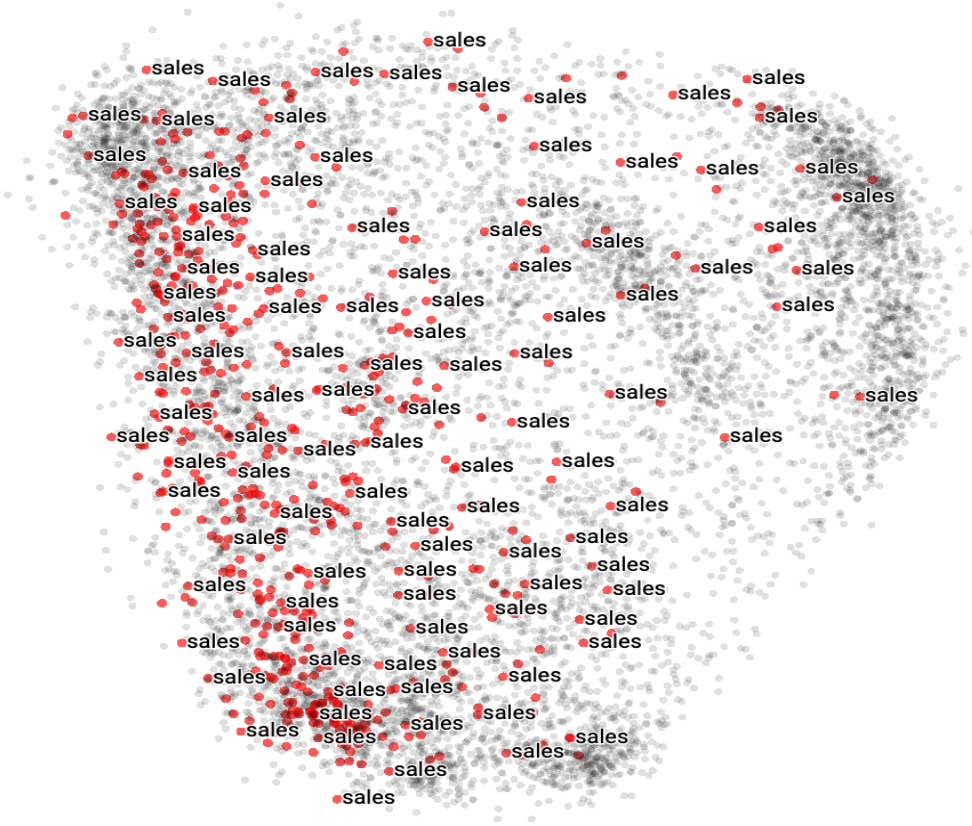
Secondly, using our model we inferred vectors all documents in the test corpus and we plot the vectors using tensor flow project to see how this documents group with each other

Some examples:

Job description with healthcare category

Job descriptions with computer_internet category

Job descriptions with sales category: are seen to be slightly more scattered



Job descriptions with administrative category: are shown to be even more sparse

We expected the group clustering results of our model to be poor in one because of the imbalance class in our dataset. However, for category (class) administrative we think the sparsity occurred because various industries require different skill sets for administrative positions. Hence, the vectors in our test dataset for administrative categories are not always close to each other.

Additionally, we noticed that in most job postings companies include information about the company, and benefits details which are not related to skills and add garbage to job_descriptions text corpus.
The other thing that we think can improve our overall performance is extracting skills keywords from documents

Extracting Skills from job descriptions using TF-IDF

While TF-IDF did not work out the way we intended it to for creating matched pairs between both the resume and job description datasets, we thought it could serve us better to be utilized as a skill extraction tool where we focus *solely* on the job description dataset. This way

we were able to easily measure evaluation metrics, as we have category labels for each of the job description records to use as our ground truth labels.

Our primary goal in this exercise was to identify and isolate key words that represent the required skills in a job description document. For this to work well, we knew we needed to eliminate as many “garbage” words as possible from job descriptions and focus on the key words.

We removed common english stop words, as well as creating a custom stoplist file of the following words that we noted were commonly used throughout all job description types:

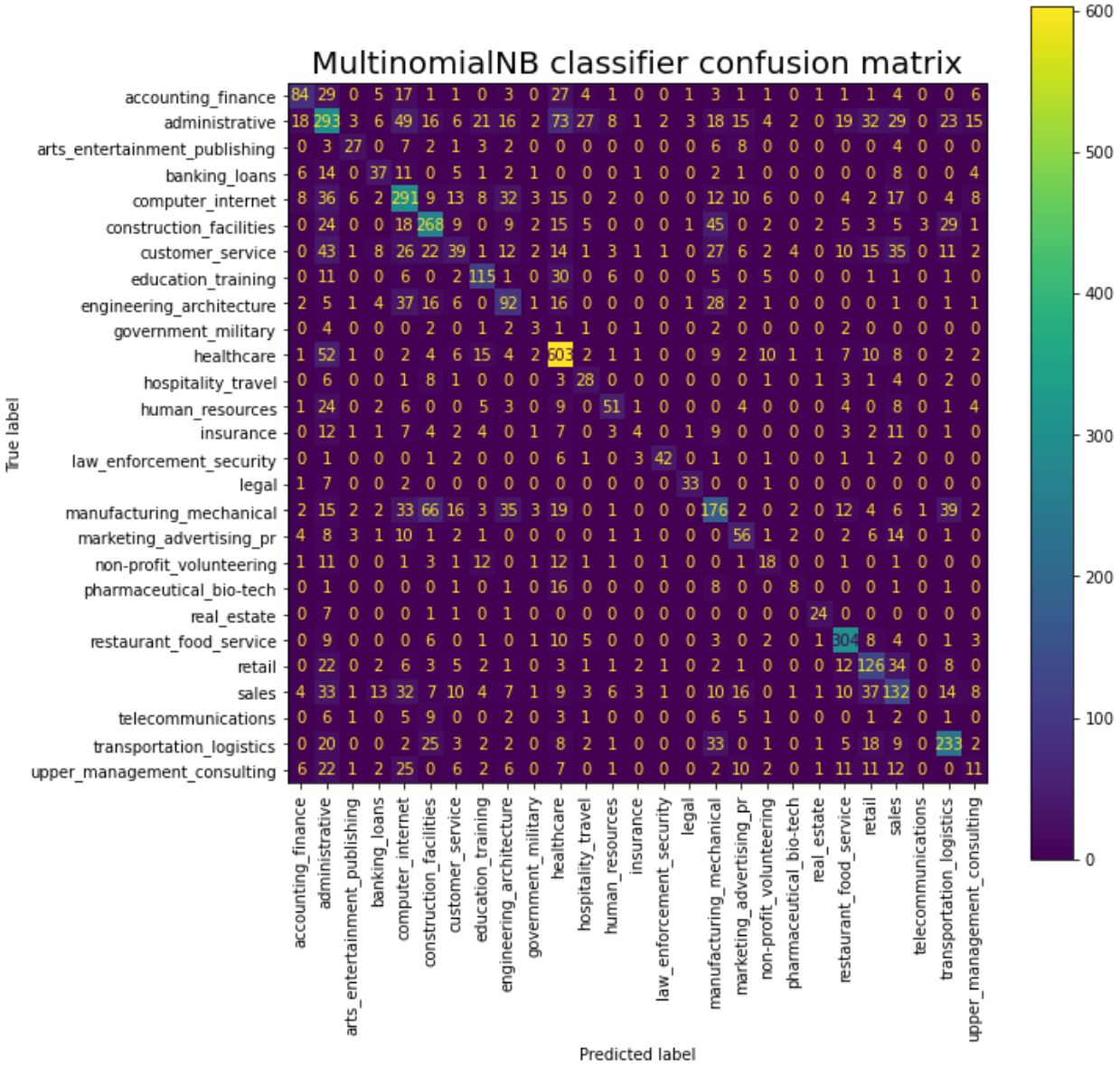
```
['business', 'work','experience','customer','work','company','technique',
'requirement', 'candidate', 'skill', 'skills','language', 'menu','program','plus',
'technology', 'job', 'technology', 'organization', 'position','required','data',
'service','location', 'type', 'ensure', 'employee','revenue','strong', 'team',
'support', 'provide', 'process', 'including']
```

To extract skills from the documents we used Multinomial Naive Bayes classifier and TFIDFvectorizer:

Our Multinomial Naive Bayes classifier has an accuracy of .52

	precision	recall	f1-score	support
accounting_finance	0.61	0.44	0.51	191
administrative	0.41	0.42	0.41	701
arts_entertainment_publishing	0.56	0.43	0.49	63
banking_loans	0.44	0.40	0.42	93
computer_internet	0.49	0.60	0.54	488
construction_facilities	0.57	0.60	0.58	446
customer_service	0.28	0.14	0.18	286
education_training	0.57	0.62	0.60	184
engineering_architecture	0.39	0.43	0.41	215
government_military	0.13	0.16	0.14	19
healthcare	0.67	0.81	0.73	746
hospitality_travel	0.34	0.47	0.40	59
human_resources	0.59	0.41	0.49	123
insurance	0.21	0.05	0.09	73
law_enforcement_security	0.88	0.68	0.76	62
legal	0.82	0.75	0.79	44
manufacturing_mechanical	0.43	0.40	0.42	441
marketing_advertising_pr	0.40	0.49	0.44	114
non-profit_volunteering	0.31	0.27	0.29	66
pharmaceutical_bio-tech	0.40	0.22	0.28	37
real_estate	0.73	0.71	0.72	34
restaurant_food_service	0.73	0.85	0.79	358
retail	0.45	0.54	0.49	232
sales	0.38	0.36	0.37	363
telecommunications	0.00	0.00	0.00	43
transportation_logistics	0.62	0.63	0.63	367
upper_management_consulting	0.16	0.08	0.11	138
accuracy			0.52	5986
macro avg	0.47	0.44	0.45	5986

```
weighted avg       0.50      0.52      0.50      5986
```



After tokenizing and building vocabulary of our corpus using TF-IDF, we built a feature array by getting feature names from the vectorizer. We then tried to build class probability indices for each of our 27 classes by taking the empirical log probability of that given class. Then we built a skills list of words by using the probability indices to take from the feature array.

We then took the top 50 adjectives from the skills list of each class.

However, we encountered a challenge related to the inclusion of irrelevant words in the resulting skills lists. Upon closer examination, we observed that certain words, such as "sexual," "dental," or "oral," appeared in the extracted skills despite their lack of direct relevance to the intended skill category.

Upon further investigation, we discovered that these words were likely seeping through from the job description sections that addressed non-discrimination policies or included information about additional benefits (such as “dental” care). While these words existed within the job descriptions, they were not of importance to us when performing identification of specific skills associated with the job categories.

The presence of these garbage words in the skills lists highlights the need for more context-aware techniques in skill extraction. This is what TF-IDF lacks, our current approach did not explicitly consider the semantic or contextual relevance of the extracted terms. As a result, some words that were not actual skills inadvertently made their way into the final lists:

*** legal *** **(the words dental, oral, new are meaningless)**
['legal', 'preferred', 'administrative', **'excellent'**, **'monday'**, **'friday'**, 'real', 'criminal', 'salary', 'corporate', 'civil', 'general', **'dental'**, 'professional', 'federal', 'associate', **'high'**, **'new'**, 'commercial', 'requirement', 'public', 'covid', 'individual', 'bachelor', 'medical', 'deadline', 'organizational', 'personal', 'degree', 'manage', 'transactional', 'help', 'global', 'good', 'parttime', 'equivalent', 'able', 'flexible', 'spanish', 'paralegallegal', 'complex', **'oral'**, 'analytical', 'successful', 'supplemental', 'assigned', 'equal', 'regulatory', 'essential', 'competitive']

** banking_loans ***(**the words sexual, dental, are meaningless)**
['financial', 'teller', 'preferred', 'new', 'high', 'excellent', 'equal', 'individual', 'responsible', 'essential', 'internal', 'applicant', 'able', 'referral', 'excel', 'professional', 'analytical', 'applicable', 'best', 'national', 'operational', 'equivalent', 'general', 'guideline', 'act', 'commercial', 'key', 'physical', 'complete', 'title', 'personal', 'federal', 'microsoft', 'fulltime', 'global', 'real', 'necessary', 'verbal', 'initiative', 'exceptional', 'standard', 'minimum', 'tool', **'sexual'**, **'dental'**, 'regulatory', 'senior', 'current', 'external', 'multiple']

*** computer_internet ***(**the words sexual, equal, new are meaningless)**
['technical', 'new', 'degree', 'global', 'complex', 'agile', 'best', 'professional', 'individual', 'responsible', 'financial', 'applicant', **'equal'**, 'excellent', 'standard', 'microsoft', 'internal', 'senior', 'implement', 'digital', 'veteran', 'expertise', 'functional', 'national', 'python', 'able', 'key', 'medical', 'federal', 'high', 'diverse', 'expert', 'strategic', 'relevant', 'equivalent', 'linux', **'sexual'**, 'collaborate', 'operational', 'learn', 'initiative', 'join', 'external', 'apply', 'analytical', 'innovative', 'essential', 'critical', 'local', 'successful']

*** accounting_finance ***

['financial', 'accountant', 'monthly', 'payable', 'related', 'prepare', 'internal', 'patient', 'general', 'resident', 'professional', 'excellent', 'perform', 'corporate', 'receivable', 'annual', 'new', 'responsible', 'senior', 'physical', 'multiple', 'high', 'medical', 'monday', 'accurate', 'able', 'analytical', 'individual', 'federal', 'external', 'key', 'document', 'equal', 'dental', 'equivalent', 'global', 'minimum', 'applicant', 'essential', 'public', 'flexible', 'necessary', 'technical', 'organizational', 'quarterly', 'local', 'personal', 'operational', 'close', 'bonus'] 50

*** administrative ***

['administrative', 'assistant', 'assist', 'medical', 'excellent', 'able', 'guest', 'fulltime', 'new', 'professional', 'responsible', 'related', 'general', 'equivalent', 'dental', 'training', 'associate', 'daily', 'candidate', 'degree', 'individual', 'financial', 'friday', 'quality', 'receptionist', 'essential', 'applicant', 'appropriate', 'minimum', 'clerical', 'necessary', 'organizational', 'multiple', 'vendor', 'equal', 'diploma', 'flexible', 'physical', 'internal', 'complete', 'use', 'positive', 'communicate', 'apply', 'key', 'good', 'federal', 'human', 'verbal', 'personal'] 50

*** arts_entertainment_publishing ***

['graphic', 'creative', 'digital', 'social', 'preferred', 'visual', 'ux', 'multiple', 'new', 'benefit', 'able', 'editorial', 'professional', 'suite', 'material', 'best', 'degree', 'excellent', 'flexible', 'mobile', 'applicant', 'technical', 'individual', 'related', 'cbs',

'interface', 'feedback', 'great', 'relevant', 'high', 'equal', 'internal', 'innovative', 'local', 'real', 'microsoft', 'collaborative', 'lead', 'medical', 'ui', 'responsible', 'national', 'good', 'covid', 'public', 'grow', 'holiday', 'shoot', 'friday', 'minimum'] 50

*** construction_facilities ***

['electrical', 'preferred', 'able', 'clean', 'fulltime', 'material', 'technician', 'high', 'general', 'dental', 'friday', 'safe', 'responsible', 'lift', 'valid', 'good', 'essential', 'physical', 'complete', 'equivalent', 'basic', 'daily', 'assist', 'individual', 'applicant', 'residential', 'great', 'level', 'read', 'yard', 'professional', 'new', 'necessary', 'follow', 'industrial', 'available', 'electrician', 'technical', 'diploma', 'competitive', 'equal', 'excellent', 'medical', 'reliable', 'lb', 'matching', 'local', 'preventative', 'flexible', 'best'] 50

*** customer_service ***

['technician', 'able', 'patient', 'technical', 'representative', 'new', 'high', 'essential', 'individual', 'assist', 'responsible', 'standard', 'medical', 'equivalent', 'applicant', 'financial', 'equal', 'best', 'general', 'task', 'multiple', 'flexible', 'good', 'internal', 'appropriate', 'positive', 'retail', 'complete', 'physical', 'necessary', 'basic', 'resource', 'national', 'timely', 'additional', 'microsoft', 'great', 'qualified', 'personal', 'competitive', 'local', 'available', 'current', 'monday', 'valid', 'follow', 'friday', 'understanding', 'reasonable', 'effective'] 50

*** education_training ***

['fitness', 'patient', 'individual', 'professional', 'instructional', 'lesson', 'clinical', 'assist', 'high', 'educational', 'social', 'behavior', 'behavioral', 'monday', 'personal', 'fulltime', 'applicant', 'able', 'physical', 'positive', 'special', 'associate', 'related', 'academic', 'available', 'new', 'medical', 'standard', 'minimum', 'current', 'flexible', 'grade', 'safe', 'daily', 'participant', 'material', 'essential', 'responsible', 'national', 'dental', 'online', 'quality', 'equal', 'public', 'effective', 'communicate', 'excellent', 'contact', 'necessary', 'referral'] 50

*** engineering_architecture ***

['technical', 'electrical', 'mechanical', 'new', 'preferred', 'civil', 'candidate', 'standard', 'lead', 'material', 'professional', 'equal', 'global', 'individual', 'medical', 'complex', 'national', 'environmental', 'assist', 'responsible', 'able', 'minimum', 'multiple', 'code', 'lab', 'sexual', 'general', 'internal', 'architecture', 'relevant', 'structural', 'industrial', 'manage', 'best', 'clinical', 'essential', 'analytical', 'impact', 'diverse', 'applicable', 'basic', 'high', 'senior', 'coordinate', 'good', 'participate', 'regulatory', 'covid', 'innovative', 'federal'] 50

*** government_military ***

['lifeguard', 'applicant', 'federal', 'aquatic', 'environmental', 'related', 'enforcement', 'public', 'social', 'assessment', 'current', 'standard', 'professional', 'individual', 'preferred', 'general', 'degree', 'human', 'new', 'minimum', 'american', 'physical', 'able', 'red', 'complete', 'national', 'essential', 'assist', 'appropriate', 'medical', 'applicable', 'equal', 'excellent', 'local', 'employer', 'material', 'open', 'grant', 'necessary', 'effective', 'bachelor', 'lesson', 'qualified', 'salary', 'fulltime', 'civil', 'high', 'responsible', 'online', 'available'] 50

*** healthcare ***

['nurse', 'medical', 'clinical', 'registered', 'dental', 'current', 'professional', 'lpn', 'appropriate', 'physical', 'certified', 'individual', 'bls', 'social', 'perform', 'able', 'high', 'new', 'personal', 'assessment', 'minimum', 'parttime', 'responsible', 'available', 'daily', 'necessary', 'direct', 'flexible', 'respiratory', 'referral', 'basic', 'equal', 'essential', 'document', 'american', 'national', 'great', 'associate', 'federal', 'complete', 'primary', 'summary', 'equivalent', 'competitive', 'reimbursement', 'occupational', 'mental', 'best', 'full', 'good'] 50

*** hospitality_travel ***

['guest', 'clean', 'preferred', 'linen', 'standard', 'able', 'high', 'resident', 'agent', 'daily', 'responsible', 'equivalent', 'great', 'friendly', 'flexible', 'necessary', 'assist', 'dog', 'available', 'essential', 'previous', 'inn', 'excellent', 'andor', 'follow', 'individual', 'reliable', 'safe', 'dental', 'professional', 'positive', 'special', 'good', 'public', 'personal', 'supplemental', 'local', 'exceptional', 'courteous', 'physical', 'polish', 'summary', 'communicate', 'multiple', 'common', 'general', 'fixture', 'appropriate', 'ensuring', 'complete'] 50

*** human_resources ***

['human', 'applicant', 'new', 'preferred', 'assist', 'degree', 'excellent', 'individual', 'professional', 'best', 'generalist', 'organizational', 'responsible', 'high', 'initiative', 'social', 'internal', 'multiple', 'equal', 'open', 'appropriate', 'federal', 'contact', 'corporate', 'local', 'able', 'dental', 'lead', 'administrative', 'strategic', 'effective', 'essential', 'medical', 'salary', 'verbal', 'interpersonal', 'fair', 'potential', 'minimum', 'current', 'key', 'flexible', 'legal', 'successful', 'personal', 'national', 'equivalent', 'referral', 'future', 'understanding'] 50

*** insurance ***

['new', 'preferred', 'requirement', 'able', 'individual', 'resident', 'professional', 'excellent', 'high', 'medical', 'financial', 'flexible', 'andor', 'associate', 'responsible', 'level', 'clinical', 'licensed', 'complete', 'federal', 'representative', 'minimum', 'human', 'public', 'good', 'microsoft', 'multiple', 'potential', 'salary', 'patient', 'supplemental', 'equal', 'independent', 'appropriate', 'great', 'equivalent', 'local', 'applicable', 'national', 'receive', 'essential', 'follow', 'administrative', 'request', 'current', 'necessary', 'general', 'competitive', 'technical', 'positive']

*** law_enforcement_security ***

['universal', 'enforcement', 'license', 'high', 'able', 'equivalent', 'posse', 'professional', 'criminal', 'valid', 'armed', 'requirement', 'guest', 'police', 'public', 'military', 'available', 'local', 'phenomenal', 'medical', 'general', 'reliable', 'dental', 'holiday', 'physical', 'effective', 'suspicious', 'operate', 'individual', 'overnight', 'responsible', 'flexible', 'appropriate', 'complete', 'safe', 'multiple', 'minimum', 'assist', 'specific', 'act', 'screen', 'equal', 'friday', 'federal', 'id', 'standard', 'north', 'write', 'essential', 'great']

*** manufacturing_mechanical ***

['preferred', 'material', 'requirement', 'able', 'high', 'automotive', 'technical', 'related', 'good', 'mechanical', 'electrical', 'basic', 'responsible', 'equivalent', 'essential', 'medical', 'new', 'applicant', 'assist', 'individual', 'equal', 'minimum', 'read', 'clean', 'physical', 'complete', 'safe', 'excellent', 'supervisor', 'friday', 'lead', 'necessary', 'pm', 'associate', 'general', 'professional', 'national', 'flexible', 'mechanic', 'daily', 'multiple', 'holiday', 'patient', 'applicable', 'environmental', 'additional', 'appropriate', 'available', 'learn', 'industrial']

*** marketing_advertising_pr ***

['digital', 'social', 'new', 'creative', 'internal', 'associate', 'strategic', 'key', 'benefit', 'website', 'insight', 'global', 'professional', 'able', 'public', 'financial', 'multiple', 'responsible', 'initiative', 'excellent', 'senior', 'individual', 'external', 'donor', 'competitive', 'applicant', 'equal', 'high', 'proven', 'coordinate', 'local', 'andor', 'promotional', 'objective', 'corporate', 'great', 'retail', 'analytical', 'remote', 'learn', 'technical', 'national', 'crossfunctional', 'apply', 'awareness', 'essential', 'organizational', 'metric', 'flexible', 'equivalent']

*** non-profit_volunteering ***

['individual', 'social', 'professional', 'behavioral', 'degree', 'report', 'public', 'physical', 'special', 'positive', 'medical', 'essential', 'new', 'appropriate', 'responsible', 'complete', 'high', 'direct', 'necessary', 'mental', 'able', 'license', 'daily', 'current', 'standard', 'implement', 'applicant', 'personal', 'residential', 'equal', 'educational', 'excellent', 'patient', 'national', 'lead', 'financial', 'flexible', 'best', 'federal', 'advocate', 'safe', 'local', 'clinical', 'minimum', 'manner', 'valid', 'good', 'administrative', 'general', 'effective']

*** pharmaceutical_bio-tech ***

['pharmacist', 'patient', 'technician', 'lab', 'clinical', 'medical', 'pharmaceutical', 'preferred', 'requirement', 'appropriate', 'license', 'high', 'dental', 'physician', 'standard', 'able', 'fulltime', 'equivalent', 'new', 'basic', 'professional', 'responsible', 'federal', 'monday', 'individual', 'technical', 'excellent', 'friday', 'essential', 'memorial', 'equal', 'necessary', 'iv', 'current', 'associate', 'covid', 'physical', 'cv', 'best', 'summary', 'additional', 'applicant', 'retail', 'full', 'national', 'texas', 'good', 'complete', 'labeling', 'clinic']

*** real_estate ***

['resident', 'prospective', 'real', 'rental', 'financial', 'prospect', 'license', 'renewal', 'professional', 'high', 'tour', 'duty', 'excellent', 'daily', 'assist', 'equivalent', 'monday', 'licensecertification', 'new', 'responsible', 'assistant', 'dental', 'residential', 'supplemental', 'administrative', 'monthly', 'potential', 'current', 'onesite', 'able', 'necessary', 'related', 'perform', 'andor', 'associate', 'shift', 'local', 'prepare', 'essential', 'great', 'competitive', 'regional', 'ready', 'experienced', 'ideal', 'general', 'regular', 'minimum', 'complete', 'social']

*** upper_management_consulting ***

['financial', 'preferred', 'general', 'new', 'guest', 'responsible', 'degree', 'strategic', 'excellent', 'key', 'assist', 'individual', 'drive', 'build', 'professional', 'schedule', 'high', 'operational', 'able', 'senior', 'technical', 'best', 'executive', 'create', 'equal', 'review', 'internal', 'human', 'global', 'multiple', 'minimum', 'corporate', 'effective', 'implement', 'analytical', 'objective', 'local', 'overall', 'hr', 'patient', 'medical', 'great', 'coordinate', 'complex', 'administrative', 'center', 'federal', 'social', 'cost', 'organizational']

Limitations

While utilizing NLP techniques such as Doc2Vec and TF-IDF proved beneficial for some aspects of job description to resume matching, we came across several limitations throughout the project. These limitations impacted the overall performance and presented challenges in determining results. The key limitations were:

1. **Lack of Ground Truth Labels:** One of the major obstacles we faced was the absence of readily available ground truth labels that could be used to evaluate the performance of our matching algorithms. This really prevented us from conducting comprehensive quantitative evaluations (collecting evaluation metrics) and limited the accuracy assessment of our techniques.
2. **Dataset Incompatibility:** We encountered difficulties in merging and aligning two separate datasets (resume dataset and the job description dataset). Due to the lack of shared ground truth labels, integrating these datasets to create a unified subcorpus for training and evaluation purposes proved very difficult. This prevented us from the construction of a more comprehensive and representative dataset for our experiments.
3. **Variability in Job Description Length**: The job description texts were immensely variable in their lengths. Some job descriptions were lengthy and detailed, while others were relatively short. This variation in text length posed issues when comparing and measuring the similarity between documents.
4. **Class Imbalance:** The distribution of job categories within the dataset was highly imbalanced. Certain categories, such as "computer_internet," had a substantial number of examples, while others, like "legal," had significantly fewer instances. This class

imbalance introduced biases that impacted the performance of our models, as they were more likely to be biased towards the majority classes, potentially overlooking important patterns and insight from the minority classes.

5. **Team's Lack of Experience in NLP (especially with Data Annotation):** Data annotation requires a deep understanding of the domain and the specific task at hand. The process of manually annotating a subset of matched pairs to create ground truth labels proved to be much more complicated and time-consuming than initially anticipated. We are sure we made plenty of mistakes when attempting to identify and label relevant information. We also did not ensure consistency and accuracy throughout the annotation process. This was due to our team's limited prior experience in data annotation for NLP tasks.

Conclusion

In this project, we investigated three main approaches: Naive Word Matching, TF-IDF Matching, and Doc2Vec-based matching techniques. Additionally, we attempted to enhance our model's performance by using TF-IDF in combination with a MultinomialNB classifier to perform skill extraction on our job descriptions dataset.

From our experimentation, we gained valuable insight into each technique. Naive Word Matching provided us with a more simple approach, while TF-IDF gave us the ability to consider term importance. Finally, Doc2Vec gave us a way to vectorize documents where we could capture semantic similarities between job descriptions and resumes. We concluded the project with an exercise on feature extraction, where we used TF-IDF and the MultinomialNB classifier to extract relevant skills from job descriptions to and make our analysis on.

Our project was not without its limitations! A big challenge was due to the lack of ground truth labels and the difficulties in merging two separate datasets together. Our limited knowledge in NLP and data annotation kept us from generating any usable, manually annotated data. Class imbalance, varying job description lengths, and disparities in category sizes also affected our implementations and final Doc2Vec model's performance, as well as what evaluations we were able to gain from our implementations.

To conclude, we explored various approaches to the problem of resume and job description matching and experimented with feature extraction on job descriptions. While our results may not have surpassed baseline approaches due to inherent limitations, we are optimistic that the lessons learned from our experiences in this project, (including, but not limited to the challenges of data annotation), will guide us towards more robust and reliable methods for job description matching and related applications in talent acquisition in the future.

References

Dutta, M. (2022, August 2). *Word2vec for word embeddings -A beginner's guide*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/07/word2vec-for-word-embeddings-a-beginners-guide/

D'Agostino, A. (2023, February 6). *How to train a word2vec model from scratch with Gensim*. Medium. https://towardsdatascience.com/how-to-train-a-word2vec-model-from-scratch-with-gensim-c457d587e031

Free resumes and job categorization dataset: Surge Ai. Free Resumes and Job Categorization Dataset | Surge AI. (n.d.). https://www.surgehq.ai/datasets/resumes-and-job-categorization-dataset

Indeed USA job listing data - Dataset by Promptcloud. data.world. (2022, March 23). https://data.world/promptcloud/indeed-usa-job-listing-data

Gensim: Topic modelling for humans. Doc2Vec Model - gensim. (2022, December 21). https://radimrehurek.com/gensim/auto_examples/tutorials/run_doc2vec_lee.html#sphx-glr-auto-examples-tutorials-run-doc2vec-lee-py

Sharaki, O. (2020, July 10). *Detecting document similarity with doc2vec*. Medium. https://towardsdatascience.com/detecting-document-similarity-with-doc2vec-f8289a9a7db7

Shperber, G. (2019, November 5). *A gentle introduction to doc2vec*. Medium. https://medium.com/wisio/a-gentle-introduction-to-doc2vec-db3e8c0cce5e