Candogram 1A: Making the job search easier!

Fall 2023, AI Studio Project Write-Up[1]

Table of Contents

Business Focus

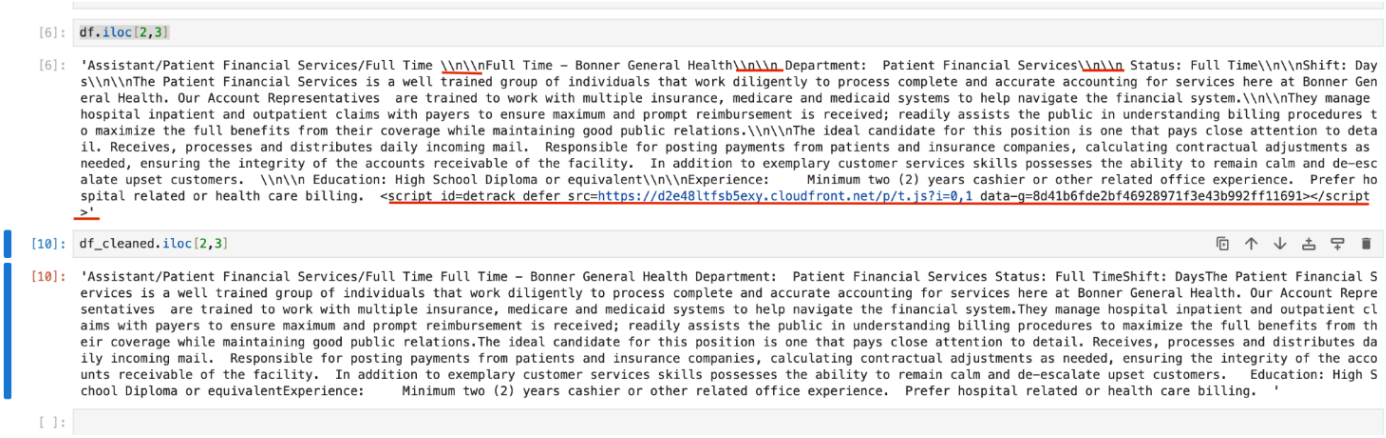
Candogram is an education-based company that focuses on high schoolers to educate them about the job market. The goal of this project was to take a dataset of 18,000+ job postings with ambiguous or fragmented titles and cluster the job descriptions based on their activities to make the job titles less ambiguous. Our team was responsible for working on jobs with “Assistant” roles. This project is important to our AI Studio host as it is a real challenge to improve the current methods Candogram uses for job title categorization. Currently, the company only uses the actual title of the job. We aim to include the job description within the data for more accurate job categorization. This project is also helpful to our host company as it will provide a more detailed job description, allowing students to make more informed decisions about their future careers.

Our Approach:

1. Data Preparation.
2. Isolating Descriptions from Job Postings using a Large Language Model.
3. Job Description Classification
4. Clustering/Topic Modeling

Data Preparation and Validation

We were initially given a CSV file of **26644** assistant job postings. For Data Cleaning, we removed HTML, hyperlinks, and unnecessary characters from job descriptions because they do not provide predictive value. We used the Python Library, **Beautifulsoup**, to achieve this output.



Our data originally had 4 columns:

- ‘Requisition ID’: The ID of the original job posting.
- ‘OrigJobTitle’: The original job title from the job posting.
- ‘JobTitle’: The general title assigned to all rows that we’ll use for clustering.
- ‘JobDescription’: The description of the job title from the original posting. The main column that we’ll be using to cluster our data.

RequisitionID	OrigJobTitle	JobTitle	JobDescription
0000JBBD	Assistant Vice President Risk Analytics and Modelling	ASSISTANT (all other)	liance\n\n**Primary Location:** North America-United States-New York-New York\n\n**Req ID:** 0000JBBD
0000JBW8	Assistant Vice President Data Scientist, Credit Risk Modeler	ASSISTANT (all other)	liance\n\n**Primary Location:** North America-United States-New York-New York\n\n**Req ID:** 0000JBW8
0000JC5X	Assistant Vice President Credit Manager	ASSISTANT (all other)	ica-United States-New York-Depew, North America-United States-Illinois-Chicago\n\n**Req ID:** 0000JC5X
00031228	HESS Assistant	ASSISTANT (all other)	-language: EN-US; > Oxy is an Equal Opportunity Employer M/F/Disability/Veteran</p>
00070-0012720350	Assistant Controller	ASSISTANT (all other)	tps://d2e48ltfsb5exy.cloudfront.net/p/t.js?i=0,1 data-g=c135f528daa6474eab77ab5f5e0d7ec61691></script>
0007v	Server Assistant/Busser/Host	ASSISTANT (all other)	tps://d2e48ltfsb5exy.cloudfront.net/p/t.js?i=0,1 data-g=3a9183d5248d465fa87204f82bcd26b91691></script>
00080-0012769939	CPA Assistant	ASSISTANT (all other)	licking Apply Now, you're agreeing to Robert Half's Terms of Use (https://www.roberthalf.com/terms-of-use) .
00090-0012747917	Assistant Controller	ASSISTANT (all other)	tps://d2e48ltfsb5exy.cloudfront.net/p/t.js?i=0,1 data-g=c5a8353bbdf74b6d8fa626e3aa03bed11691></script>
00090-0012764427	Accounts Payable Assistant	ASSISTANT (all other)	licking Apply Now, you're agreeing to Robert Half's Terms of Use (https://www.roberthalf.com/terms-of-use) .

Feature Selection

Since the goal of our project focuses on working with Job Descriptions, for feature selection we decided to remove the columns that weren’t necessarily important or related to our goal. We removed the columns: OrigJobTitle and JobTitle and reduced our CSV file to two columns: RequisitionID and JobDescription.

Since we are only working with Assistant roles, every row consists of “Assistant”. Therefore, the column JobTitle provides no predictive value.

Since we will be assigning new jobs based on the job descriptions, the column OrigJobTitle wasn’t necessary anymore.

We decided to keep the column RequisitionID to keep track of where we received the data from. And the column JobDescription since we need to use it for our predictions.

	RequisitionID	OrigJobTitle	JobTitle	JobDescription
	0000JBBD	Assistant Vice President Risk Analytics and Modelling	ASSISTANT (all other)	liance\n\n**Primary Location:** North America-United States-New York-New York\n\n**Req ID:** 0000JBBD
	0000JBW8	Assistant Vice President Data Scientist, Credit Risk Modeler	ASSISTANT (all other)	liance\n\n**Primary Location:** North America-United States-New York-New York\n\n**Req ID:** 0000JBW8
	0000JC5X	Assistant Vice President Credit Manager	ASSISTANT (all other)	ica-United States-New York-Depew, North America-United States-Illinois-Chicago\n\n**Req ID:** 0000JC5X
	00031228	HESS Assistant	ASSISTANT (all other)	-language: EN-US; > Oxy is an Equal Opportunity Employer M/F/Disability/Veteran</p>
	00070-0012720350	Assistant Controller	ASSISTANT (all other)	tps://d2e48ltfsb5exy.cloudfront.net/p/t.js?i=0,1 data-g=c135f528daa6474eab77ab5f5e0d7ec61691></script>
	0007v	Server Assistant/Busser/Host	ASSISTANT (all other)	ps://d2e48ltfsb5exy.cloudfront.net/p/t.js?i=0,1 data-g=3a9183d5248d465fa87204f82bcd26b91691></script>
	00080-0012769939	CPA Assistant	ASSISTANT (all other)	licking Apply Now, you're agreeing to Robert Half's Terms of Use (https://www.roberthalf.com/terms-of-use) .
	00090-0012747917	Assistant Controller	ASSISTANT (all other)	tps://d2e48ltfsb5exy.cloudfront.net/p/t.js?i=0,1 data-g=c5a8353bbdf74b6d8fa626e3aa03bed11691></script>
	00090-0012764427	Accounts Payable Assistant	ASSISTANT (all other)	licking Apply Now, you're agreeing to Robert Half's Terms of Use (https://www.roberthalf.com/terms-of-use) .

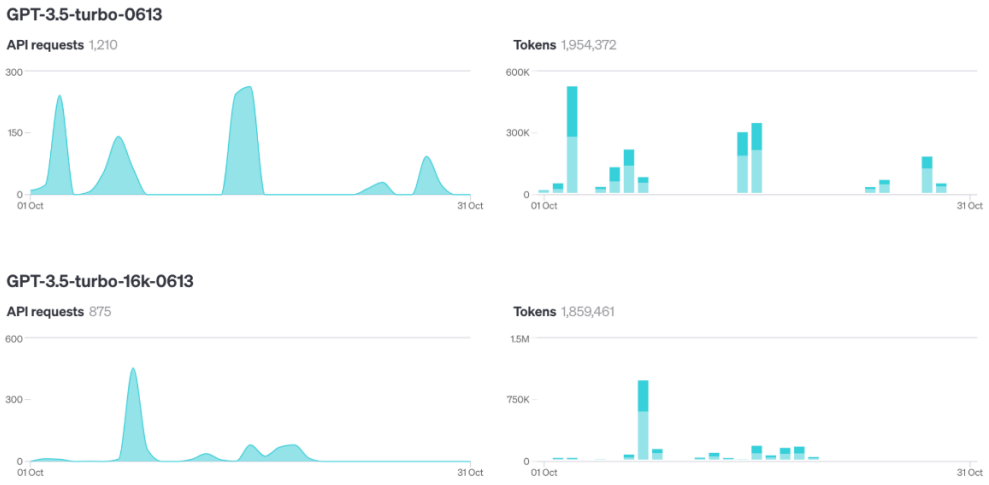
Approach

LARGE LANGUAGE MODEL FOR TEXT PREPROCESSING

- For our large language model, we used Open AI's AGI(Artificial General Intelligence) tool called "**gpt-3.5-turbo-16k."**
    - It's an improved version of GPT-3, which ChatGPT used when it was released. Its training data holds up to September 2021; an individual can input 16,385 tokens to generate an output.
    - For reference, the newest GPT-4 model uses training data up to April 2023 and allows the user 128,000 tokens per input.
- Pricing:
    - Our stakeholder gave us a 60 dollar limit to generate our output
    - 1 token is 4 characters.

Model	Input	Output
gpt-3.5-turbo-1106	$0.0010 / 1K tokens	$0.0020 / 1K tokens
gpt-3.5-turbo-instruct	$0.0015 / 1K tokens	$0.0020 / 1K tokens

The following chart gives an idea of our API Requests and token usage over the month of October.



We used this API to classify each sentence from a job description into 4 categories:

- '**Marketing**': Information about the company, promotional opportunities, or any marketing-related aspects
- '**Description'**: Tasks one will do for the role, responsibilities, and nature of the role.
- '**Requirement'**: Qualifications, skills, and prerequisites necessary to apply for the job.
- '**Legal**:' Legal information, compliance requirements, etc.

We defined these categories as '**Labels'** and created a prompt for the large language model (LLM) to assign specific sentences from the job description into the 4 categories. Every team member made similar but different prompts and received the same output.

Here are examples of the prompts used:

1. Initiating data input for job post classification. Each sentence in the job posting requires categorization into predefined groups: 'Marketing,' 'Description,' 'Requirement,' or 'Legal.'

   'Marketing': Assign sentences conveying information on the company, promotional opportunities, or marketing-related aspects of the role.

   'Description': Allocate to sentences detailing tasks, responsibilities, and the overall nature of the position.

   'Requirement': Apply to sentences enumerating qualifications, skills, and prerequisites necessary for job application.

   'Legal': Designate sentences containing legal information, including compliance requirements in the job description.

   Ensure classification based on sentence content, with separation denoted by a period(.), question mark(?), or exclamation mark(!).

   Present the response in valid JSON format, structured as "sentences," then "sentence," and "category."

2. You are given a job posting. Break up the job posting into sentences. Create a JSON object with the categories "Marketing", "Description", "Requirements", "Legal".

   Marketing: Single text section where the company describes itself. Select sentences about the company marketing into the "Marketing" category.

   Description: Single text section where the company describes the job activities. Select sentences about the job description into the "Description" category.

   Requirements: Several text sections about requirements with preferred qualifications, experience, skills, knowledge, abilities, education, degrees, competencies, credentials, bachelor requirements. Select sentences about job requirements into the "Requirements" category.

   Legal: Single text section where the company speaks about its legal work policies. Select sentences about company legal terms into the "Legal" category.

   For each category place sentences into an array. Return a RFC8259 compliant JSON response.

Here's the Code used to create the response and save it in a new csv file that would be later used for text classification.

```
with open(output_file, 'w', encoding='utf-8') as outfile:
   writer = csv.writer(outfile, quoting=csv.QUOTE_ALL, lineterminator='\n')

   for idx, row in job_descriptions.iterrows():
       if len(row['JobDescription']) < 4000:
           job_description = row['JobDescription']
           req_id = row['RequisitionID']
           #print (len(row['JobDescription']))


           response = openai.ChatCompletion.create(model="gpt-3.5-turbo",
                                                   messages=[{"role": "user", "content": f"""{prompt}
                                                   ----------------------------------------
                                                   job posting:{job_description}"""}])

           json_obj = json.loads(str(response["choices"][0]['message']['content']))
           #print (json_obj)

           if 'Description' in json_obj.keys():
               for row in json_obj['Description']:
                   #print (row)
                   writer.writerow([req_id, "Description", row ])

           if 'Marketing' in json_obj.keys():
               for row in json_obj['Marketing']:
                   #print (row)
                   writer.writerow([req_id, "Marketing", row ])

           if 'Requirements' in json_obj.keys():
               for row in json_obj['Requirements']:
                   #print (row)
                   writer.writerow([req_id, "Requirements", row ])

           if 'Legal' in json_obj.keys():
               for row in json_obj['Legal']:
                   #print (row)
                   writer.writerow([req_id, "Legal", row ])
```

Our previous file only had the columns 'Requisition ID', and 'JobDescription'. But the new file with data from the preprocessing had the columns, 'Requisition ID', 'Content' (each sentence separated from the job description), and 'Label' (tells us whether it is a description, marketing, legal, or requirements sentence).

Before:

ReqID	Job Description
0000JC5X	ıcations: North America-United States-New York-New York, North America-United States-New York-Depew, North America-United States-Illinois-Chicago Req ID: 0000JC5X

After:

ReqID	Content	Label
0000JC5X	Credit Manager position will cover a diverse portfolio of U.S. headquartered clients.	description
0000JC5X	C products and services within Commercial Banking and the broader organization.	description
0000JC5X	ər, you will proactively monitor the ongoing credit quality of your assigned portfolio.	description
0000JC5X	s, tracking and testing covenants, assisting with credit reviews, internal audits, etc.	description
0000JC5X	rs, our internal Legal/Compliance team, to complete the process from start to end.	description
0000JC5X	ıt, you will also recommend appropriate risk ratings for each customer/transaction.	description
0000JC5X	ınd variable pay, as part of an employee's overall total compensation and benefits.	marketing
0000JC5X	ıkes the form of discretionary, annual awards (sometimes referred to as a bonus).	marketing
0000JC5X	s designed to help you improve your health and well-being, finances, and lifestyle.	marketing
0000JC5X	HSBC is committed to employing only those who are authorized to work in the US.	legal
0000JC5X	k in the U.S. as HSBC will not engage in immigration sponsorship for this position.	legal

TEXT CLASSIFICATION: USING THE FASTTEXT MODEL

Next, we move on to text classification. Our objective for this part of the project is to develop a classifier that will accurately extract job description sentences from the remaining job postings using the labeled data from the LLM.

To achieve this, we chose to use **fastText**, which is a powerful text classification tool developed by Facebook's AI Research (FAIR) lab that efficiently categorizes text data based on predefined labels.

This pre-trained language model transforms sentences into embeddings, which we then use to classify the sentences from the remaining job postings.

DATA PREPROCESSING

A fastText model has some requirements to be met regarding its input:

It requires a labeled dataset for training, where each text is associated with a predefined label. It also accepts text in plain format, with labels prefixed by the word "label".

For enhanced classification, we did some data preprocessing. Since we are only focusing on the job descriptions for this project, we updated the 'description' label to 'Label_1', and the rest of the labels to 'Label_2'. We then saved the updated data in a text file.

Before:

FastText.ipynb | job_sentences.csv | job_sentences_train.txt

Delimiter: ,

	Label	Sentences
23	Description	Contacts: Surgical team including surgeons, anesthesiologists, nurses, scrub techs and perfusionists.
24	Description	Some interaction with other health care providers.
25	Description	Minimal verbal interaction with patient and families.
26	Marketing	Assist the surgeon during surgery.
27	Marketing	Ensure the availability and functioning of surgical equipment used in surgical procedures.
28	Marketing	Transport patient to and from operating room, preps patient prior to surgery, serves as first or second assistant to the surgeon during the surgical procedure.
29	Requirements	st meet one of the following: RN who has completed a Surgical First Assist program (certification is preferred) or Nurse Practitioner who has completed a Surgical First Assist program.
30	Requirements	Experience: Must be certified tech, RN or NP and must be able to perform Endoscopic Vein Harvest.

After:

job_sentences.csv | job_sentences_train.txt

```
Label_1,"Contacts: Surgical team including surgeons, anesthesiologists, nurses, scrub techs and perfusionists."
Label_1,Some interaction with other health care providers.
Label_1,Minimal verbal interaction with patient and families.
Label_2,Assist the surgeon during surgery.
Label_2,Ensure the availability and functioning of surgical equipment used in surgical procedures.
Label_2,"Transport patient to and from operating room, preps patient prior to surgery, serves as first or second assistant to the surgeon during the surgical procedure."
Label_2,Education: Must meet one of the following: RN who has completed a Surgical First Assist program (certification is preferred) or Nurse Practitioner who has completed a Surgical First Assist program.
Label_2,"Experience: Must be certified tech, RN or NP and must be able to perform Endoscopic Vein Harvest."
```

As you can see, the CSV file had different labels, description, marketing, and requirement. Later we saved it as a text file, and it only has label_1 and label_2.

DATA SPLITTING

Next, we perform Data splitting where we split the sentences and labels into training and testing sets.

The training data is the one that's used to train the model. The model observes and learns from the training set and optimizes its parameters.

After the training phase is complete, we use the testing set to understand the performance of the model in comparison to different models and hyperparameter choices. We use the Python library **scikit learn** to achieve this.

For increased model compatibility, we used the function "**fillna**" to remove the potential NaN values from the data.

To meet the input requirements of the fastText model as stated before, we prepared the data by including label prefixes and concatenation. We then wrote the training and testing sets into plain text files.

test_fasttext.txt

```
__label__Label_2 We foster a work environment that feels like a family and have a culture of compassion.
__label__Label_2 Pediatric Specialty Care is dedicated to meeting the unique needs of medically complex and technology dependent individuals ranging from birth to 21 years of age.
__label__Label_2 In addition, we maintain an abundant supply of PPE, including N95/KN95 masks, for all who provide care and services to our patients and residents.
__label__Label_1 As the Assistant Controller, you will assist in supervising the accounting staff and report to the Controller.
__label__Label_2 We Develop YOU! We provide career ladders, education and training opportunities so you can build a long and successful career with Genesis.
__label__Label_2 Apply today.
```

train_fasttext.txt

```
__label__Label_1 Co-constructing curriculum with the head counselor, based on children's interests.
__label__Label_2 Possess the ability to make independent decisions when circumstances warrant such action
__label__Label_2 Candidates must pass required state and company background checks, and meet state and company minimum education and experience requirements:
__label__Label_2 We will provide all the resources you will need to design your fun filled days!
__label__Label_1 Activities Assistant: * Assist with planning and organizing activities that provide opportunities for entertainment, exercise, relaxation, and expression to our Residents.
__label__Label_2 People support your growth here and there's a team spirit that makes the tough days a whole lot easier.
__label__Label_2 Central Desert Behavioral Health Hospital is a 64-bed inpatient hospital offering advanced programming and services for the treatment of behavioral health illness for older adults.
```

MODEL TRAINING AND EVALUATION

The fastText model's accuracy is influenced by various hyperparameters, including:

epoch: The number of times the training data is passed through the model.

wordNgrams: The maximum length of word n-grams considered during training.

lr: The learning rate, which determines the step size during optimization.

dim: The dimensionality of the word vectors.

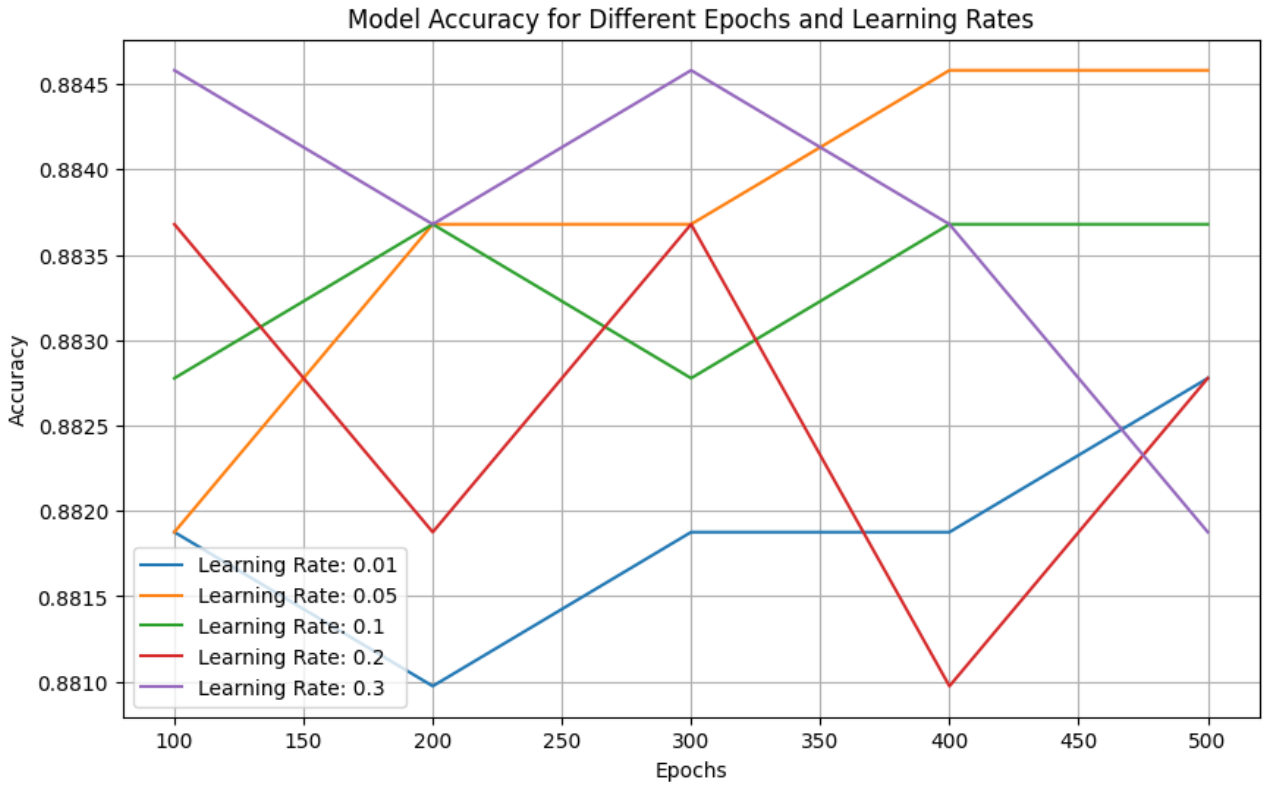
We train a fastText model by specifying hyperparameters, and tuning these hyperparameters can significantly impact the model's performance.

Each of us tried experimenting with different hyperparameters to see which gives the highest accuracy, and achieved the following results:

Epoch: 200, lr: 0.01, word N-grams: 2, dim: 200 => Achieved accuracy 90.6%

Epoch: 500, lr: 0.01, word N-grams: 2, dim: 200 => Achieved accuracy 89.96%

Epoch: 300, lr: 0.30, word N-grams: 2, dim: 200 => Achieved accuracy 88.5%



This plot shows how changing hyperparameters like epochs and learning rates can make the accuracy of the model significantly vary.

TEXT CLASSIFICATION

For text classification, we first extracted Requisition ID and Job Descriptions from the remaining job postings.

Then we Tokenized the job descriptions into individual sentences and used our pre-trained fastText model to predict the labels and confidence scores for each sentence.

Sentences labeled as '__**label**__**Label_1**' with a confidence score greater than or equal to **0.95** were considered 'description' sentences.

These 'description' sentences were then aggregated and stored in a dictionary where the keys are Requisition IDs, and the values are lists of classified descriptions.

Lastly, a new DataFrame was created with the aggregated job descriptions and saved in a CSV file.

Here is the csv file with classified job descriptions with their respective Requisition IDs.

	ReqID	**Job Description**
	JP-003772148	Enforces adherence to established policies and processes. \n\nOversees trains and manages the accounting & AP staff
1	02880-0012214770	assisting with financial statement preparation; pulling various financial reports; internal controls; other projects as assigned
2	2023-1486	ntains a positive attitude despite adversity. Builds an effective team by engaging, mentoring, and developing direct reports
3	04160-0012602062	a self-starter with a demonstrated history of identifying issues and problem solving for this exciting Assistant Controller role
4	R-59288	Managing flow of accounting data to/from remote facilities
5	02340-0012609214	This Assistant Controller role reports to the Controller and manages the accounting staff
6	02940-0012597820	\n\nThe CFO is looking for an exceptional Asst
7	01380-0012657892	ng Assistant Controller position. You will keep the Controller up to date on the accounting staff as the Assistant Controller
8	04340-0012562520	supervise the accounting staff while reporting to the Controller. Submit an application now and learn more about this role
9	R-60792	Managing flow of accounting data to/from remote facilities
10	CLIMBER	Duties Description Maintain facility security. Key control. \n\n Duties Description Maintain facility security. Key control
11	INTER006150	n and support any related tax and government filings. \n\n+ Actively support the budget process (annual and 5-year plans)
12	R-55563	\n\n+ Managing flow of accounting data to/from remote facilities
13	ASSIS001508	and vendors. \n\n+ Assist with the annual financial audit and tax filing preparation. \n\n+ Other related duties as assigned
14	00610-9503687752	10Q and Form 10K. \n\n+ Provide support for the quarterly reviews and annual audit with the Company's external auditors
15	JP-003753918	\n\nOther responsibilities as assigned
16	02890-0012615686	As the Assistant Controller, you will help oversee the day to day operations of the accounting function
17	CUSTO004084	dual. \n\nSupervisory Responsibilities/Direct Reports: \n\nThis role may serve as a team lead for an assigned work group

MODEL COMPARISON

For Model Comparison, we compared our fastText model to another pre-trained language model called **BERT**.

With BERT, we achieved a slightly higher accuracy of 91.1%. However, we decided to stick with fasText since the BERT model was taking FOREVER to run. Where the fastText model took around 2 minutes, the BERT model ran for like an hour!

We also realized that training BERT models from scratch can be more computationally costly.

Therefore, overall we decided that fastText was a better option.

Model Name	Description	Results	Pros	Cons
fastText	An efficient model for learning word representations and text classification, with a focus on sub-word information.	**90.6 % accuracy**	fastText is highly efficient. It can handle out-of-vocabulary word	Lower contextual understanding than BERT on complex NLP tasks
BERT	Pre-trained language model that excels at understanding the contextual meaning of words and sentences in natural language.	91.1% accuracy	High contextual understanding	Takes forever to run. Training BERT models from scratch is more computationally costly.

CLUSTERING / TOPIC MODELING

Now that we have isolated the job descriptions from each job posting, the goal is to combine groups of similar jobs to categorize the data.

There are two different methods to separate the data into distinct categories:

Clustering: a data analysis technique that groups similar data points based on certain features. It identifies natural grouping within a data set

Topic modeling: a subset of machine learning that aims to identify topics and themes within a collection of textual documents by uncovering hidden patterns within the text

Both of these techniques play a role in recommendation systems by grouping user data to allow for more personalized recommendations and suggestions.

DATA PREPARATION FOR CLUSTERING / TOPIC MODELING

Before running either a clustering model or a topic modeling model on the data set, we need to prepare the data for the Natural Language Processing (NLP) task by removing stop words.

These are common words that don't have inherent meaning to the text such as the, a, am, ext. Then we removed numbers and punctuation. Later we performed lemminzation which breaks down words into their root meaning. For example, running, ran, and run would all become run.

Both data sets also need to be converted into numerical values before they can be processed by a computer.

For clustering, the data is vectorized and then it is run through **TFIDF** which stands for Term Frequency inverse document frequency. This pulls out words that are important in the overall collection of data. Based on this, each job description was boiled down to its keywords

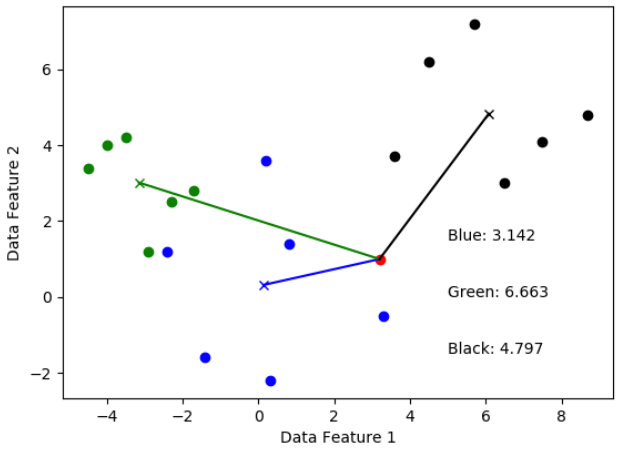
For topic modeling, a **Document Term Matrix** is used.

Once we preprocessed and vectorized the data, we performed both a clustering method and a topic modeling method to compare the results from each of them.

K-means Clustering

For clustering, we used a technique called **K-means**. It is a clustering algorithm used to group similar data points together. The algorithm groups data points, each one representing a word, into clusters by finding a central point (centroid) for each cluster. It then employs a distance-based approach to find the cluster's centers (centroids) and assign data points to the nearest centroid to determine which cluster they are in.

This method is shown in the image below



After clustering, we extracted keywords from each cluster to help identify the main topic or theme of that cluster.

Here are 5 of our Topic Results from K-means clustering:

Cluster One: veterinary, clients, description, care, safety, ensuring, quality, summary, assistant, job

Cluster Two: assistant, assist, activities, support, job, program, assigned, responsibilities, project, responsible

Cluster Three: service, providing, assist, customer, manner, responsibilities, assisting, assistant, job, role

Cluster Four: patient, patients, care, medical, performs, procedures, support, health, assists, assigned

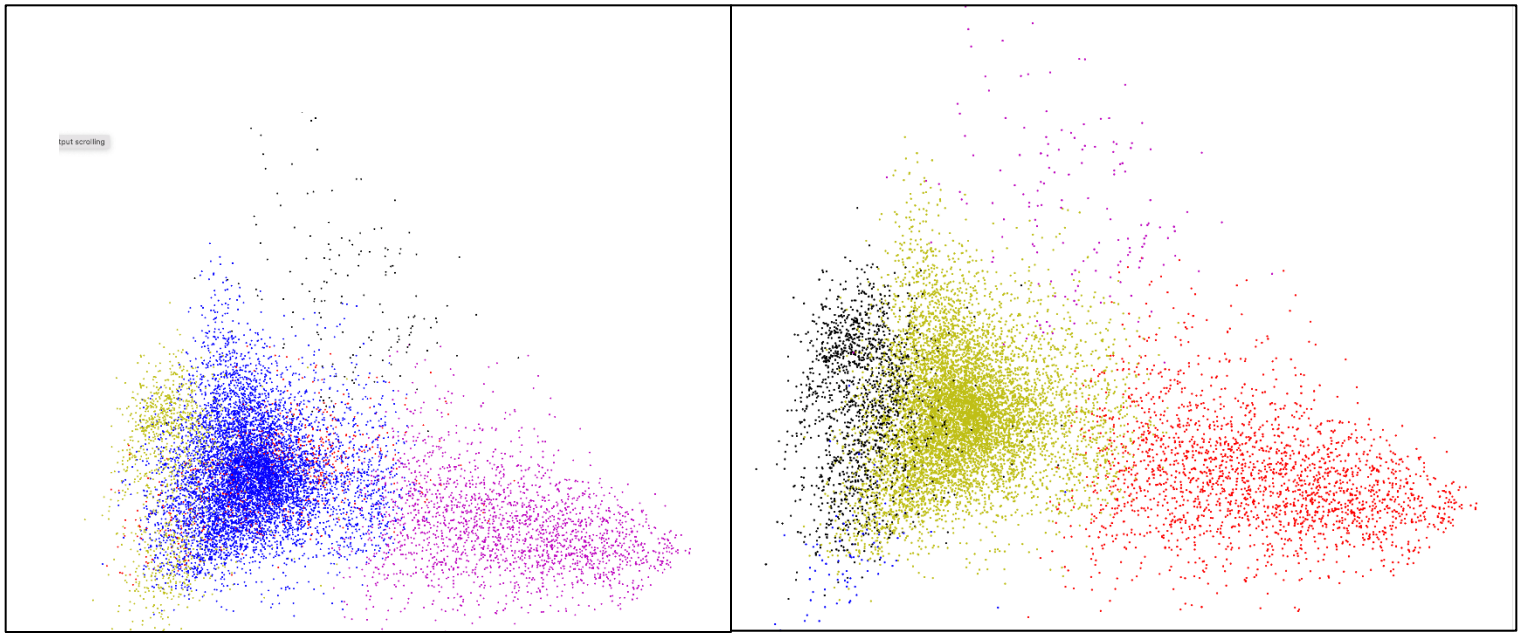
Cluster Five: manager, store, customer, leader, team, team leader, sales, assistant, assist, product

To better understand and communicate the results, we decided to visualize the clusters.

The output of the K-means algorithm is data points assigned to their respective cluster. Visualizing these groups by assigning a color to each group helps reveal patterns and insights that are not immediately obvious from the raw data. The way to determine if the clustering algorithm was successful is if there are distinct groupings and no overlap between the clusters.

Here is the K-means Visualization for iterations 100 and 500.

Iteration= 100 **Iteration=500**



As you can see, with 500 passes through the algorithm, there are distinct groupings. There is black, yellow, red, and magenta on the top, and blue in the bottom left corner.

We used this clustering method as a preliminary test to see if it was possible to cluster the data. We then turned to a more robust machine-learning approach through topic modeling using Latent Dirichlet Allocation (LDA).

Topic Modeling Technique: LDA

Latent Dirichlet Allocation or LDA is used to discover hidden topics within a collection of documents. It determines which topics are present in each document and which words are related to those topics.

The first step to doing this is by making a Document Term Matrix. It serves as a way to represent textual data in a numerical format that is suitable for computational analysis.

Each row in the dataset represents a job description and each column represents a word appearing in the job descriptions. The values are the frequency count.

Then we create a Corpus where each word is matched to a numerical ID.

The LDA algorithm assigns topics to each document and extracts keywords for these topics.

It Initializes random assignments of words to topics in each job description and then Iteratively reassigns words to topics based on continuous probability distribution calculations. This process is repeated for a set number of iterations.

Later, the LDA model infers probable topics for each job description and overarching words associated with each topic. The outcome is topics, each represented by a word distribution, and for each job description, a topic distribution.

Topics are interpreted based on words with high probabilities, allowing us to assign meaningful categories.

LDA: Hyperparameters

These are the parameters we used to accomplish a successful LDA topic distribution model. The corpus was used to assign words to numerical IDs, and id2word was used to extract the words from their numerical representation.

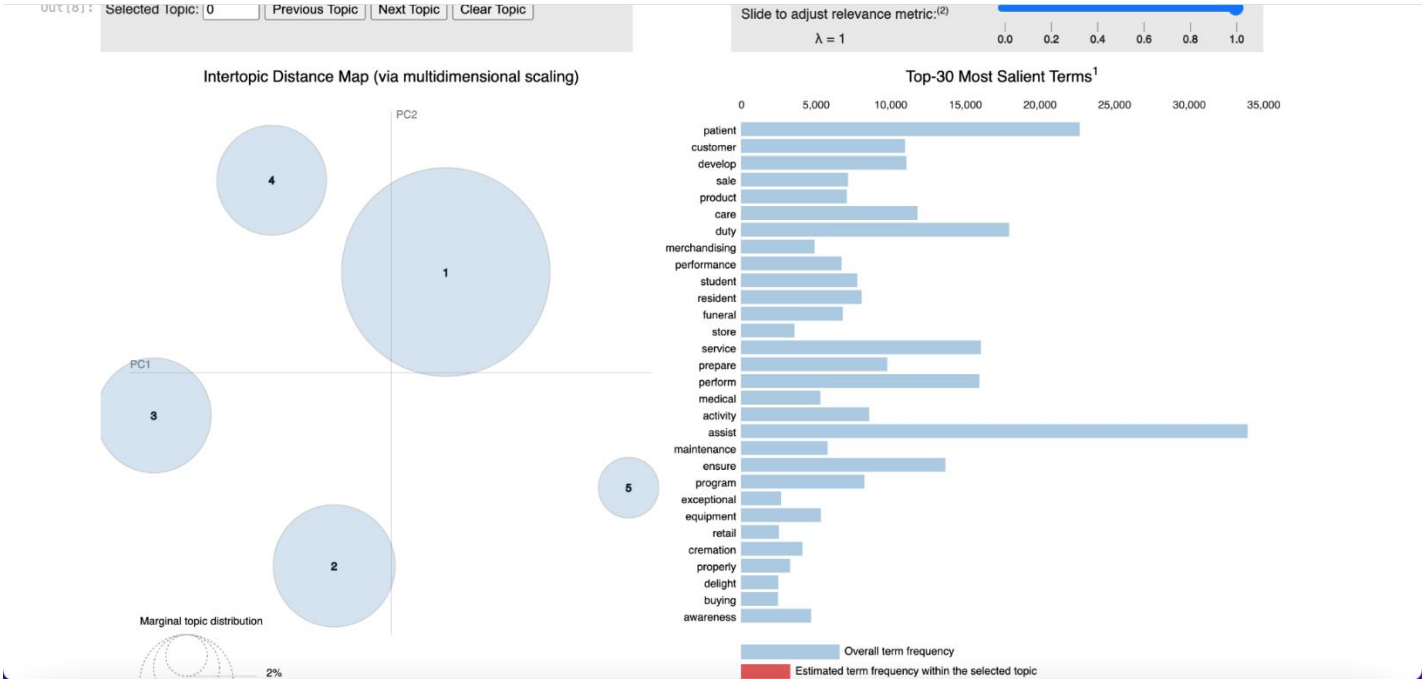
We chose five topics based on the successful k-means clustering and had it iterate through the data 20 times.

```
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                            id2word=id2word,
                                            num_topics=5,
                                            random_state=100,
                                            update_every=1,
                                            chunksize=50,
                                            passes=20,
                                            alpha="auto")
```

LDA: Visualization

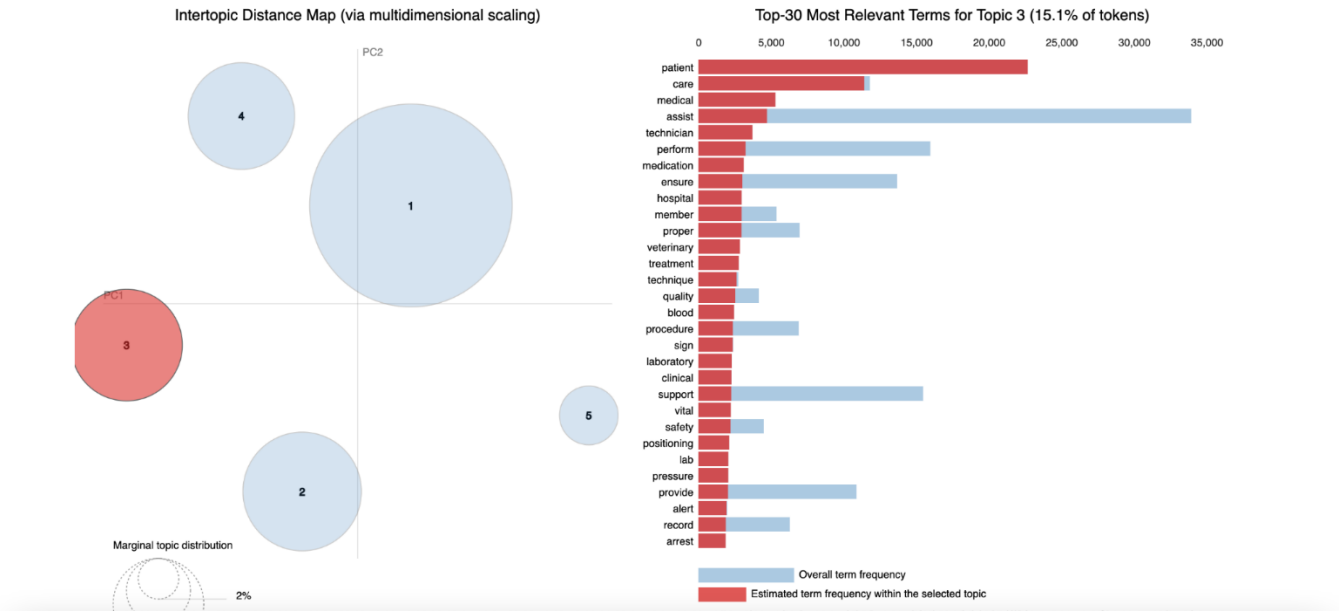
The LDA model also outputs a very nice visual representation.

Here you can see that our topics do not overlap. You can also see that the word **assist** is the most common word in the model which makes sense given that we are using the Assistant jobs database.



If you hover over a certain topic, you can see the most common terms within that topic.

For example, topic three's most common term is patient and the rest of the words are also related to medical things, making a logical category for this topic to be hospital assistant.



LDA: Topic Results

Below are the extracted words for the five topics we have. The first topic seems to be for a technician assistant, the second for retail store assistant, the third, as we mentioned, a hospital assistant, the fourth a funeral home assistant, and the fifth a university campus assistant.

Topic 1: maintain, assist, project, work, support, operation, labor, include, report

Topic 2: customer, sale, product, merchandising, develop, store, exceptional, retail, delight

Topic 3: patient, care, medical, assist, technician, perform, medication, ensure, hospital

Topic 4: duty, service, funeral, prepare, assist, perform, maintenance, equipment, cremation

Topic 5: assist, student, resident, activity, provide, program, awareness, environment, achieve

Key Findings and Insights

For Text Classification, we successfully developed a classifier that will accurately extract job description sentences from the job postings using the labeled data from the LLM. For this, we tried using 2 pre-trained language models, fastText and BERT. We achieved a 90.6% accuracy with fastText within 2 minutes and prioritized its notably quicker runtime over BERT's 91.1% accuracy.

We used the K-means clustering method as a preliminary test to see if it was possible to cluster the data. We then turned to a more robust machine-learning approach through topic modeling using Latent Dirichlet Allocation (LDA).

Based on our LDA Topic Results, we are proud that we accomplished our overall goal, i.e. showing it was possible to extract job descriptions from job postings with ambiguous or fragmented titles and categorize them into distinct topics.

We are confident that this can be fine-tuned and applied further to create distinct groupings which will allow students to make more informed decisions about their careers.

In the future, we would also like to classify the requirements section of job postings so that high schoolers know what type of skills they should focus on developing based on the career path they are interested in.

This was one of my first experiences working on a team tackling a technical project of such a large scope. The most rewarding lesson has been the power of collective problem-solving. Tackling complex technical issues led to innovative solutions that no individual could have conceived alone. This collective problem-solving approach not only enhanced the quality of the final product but also fostered a sense of camaraderie among team members. That said, our team benefited most from setting, clear and open communication, and norms. Regular team meetings, status updates, and openly sharing when we got stuck helped to ensure that everyone was on the same page.

Acknowledgments

I've learned so much in these last 10 weeks. To my Challenge Advisor Henning and TA Kimia, and teammates Michelle, and Noah, thank you for your support, and the opportunity to learn alongside you!

Additionally, thank you to Break Through Tech, the Cornell Tech AI Program team, and especially Erika and Abby for an amazing program experience so far!