Analysis of rejection and non-rejection emails of a software engineer

Introduction

Rejection emails are a very common occurrence in today's modern digital employment architecture. In this report we are looking at how rejection emails differ to other emails and if common trends are picked up. We are doing this by performing different experiments and see what we can find with the results. After performing the experiments, we can evaluate our findings and discuss the dataset used.

Initial analysis

The dataset is created by Seth Polyniak to automate responses to job rejections through detecting them with machine learning. It has 129 datapoints with 65 being rejection emails and 64 being non-rejection emails meaning it is pretty much an even split of data.

Topic modelling

It is quite apparent that there are topics that can be present throughout the dataset which can be seen through themes. Due to that, I decided to use LDA and NMF to investigate the topics found. I also wanted to use more effective models based on embeddings like Bertopic and Top2Vec but turns out they don't work on datasets this small as the code kept throwing errors based on the size of the dataset which indicated how it was struggling to find topics effectively.

Method

LDA

The data was first pre-processed by converting all text to lower, removing non-alphabetic characters and stop words. 1500 passes were made in the lda model to ensure the 12 topics are fully understood by the model to produce as good of a model as possible to identify the topics. This was applied to non-reject and reject topics separately. Stemmers muddled up the results and didn't help much in finding latent topics.

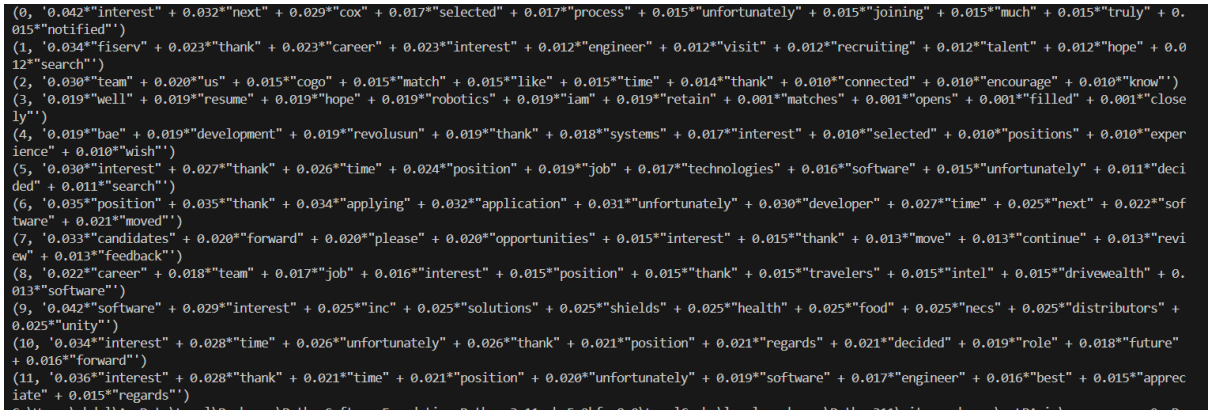
NMF

For NMF, the TFI-DF vectorizer was initialised with a max_df of 0.95 to remove terms that don't help us find a difference between the emails, a min_df of 2 so it appears in at least 2 emails and cleaned up the data inputted by removing stop words. The NMF was set to

find 12 subjects, that number was arrived since after 12 the topics found were close together and weren't as useful. This was applied to non-reject and reject separately.

Results

LDA



It highlights three extremes which show the topics that are the furthest from each other and there are some that are less extreme showing there is some common ground between the topics in the email. One of the main things I noticed was one of the topics that was found was like experience and skills of Seth like his reasoning results and bachelor's degree. The topics that are closer are services like Steam (and Nintendo inside the same topic), Amazon, MyJobHelper , Ebay. There are some topics also seemed very job focused like topics 10,7 and 1.

NMF

(Detailed results in appendices)

Non-reject

There is a lot of mentions of services and support in the non-rejection emails like seen in topic 4,5 ,10 meaning the non-rejection include the services and support that Seth has used to perform daily activities like selling items on ebay and getting ads for amazon prime as well as signing in on steam. Also, there are lots of positive topics like 8,1,9 in which there is a lots of positive words like delighted shows that they want to be seen as more engaging than rejection emails.

Reject

Phrases like "appreciate your interest" and moving forward" indicate polite rejection of the person which show higher formality. Lots of focus on the application process as

seen in topic 3 and 9. The topics and the words used are very depersonalised and sounding more neutral to show less engagement is needed from the receiver.

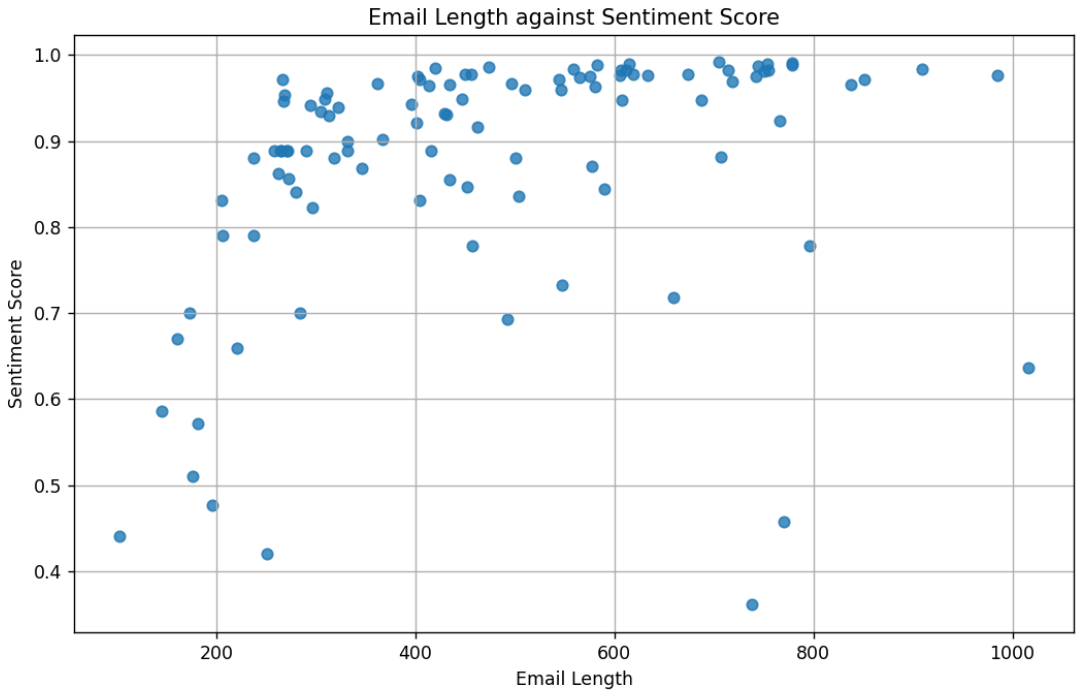
Sentiment analysis.

In my opinion I feel like all rejection have a similar sentiment and I want to see how true that is. Another interesting thing to analyse is to see if email length effects the sentiment presents in rejection emails.

Method

Scatter graph of all the sentiment scores that have been filtered using interquartile range and plotted against length. Sentiment score was calculated using Vader Intensity analyser. Scores more than .05 were seen as positive and less than .05 negative and the middle range was seen as neutral. Average of positive, negative, and neutral was found through median as it accounts for outliers while still using those values. Email length was calculated with characters.

Results



Average email length by sentiment:

sentiment

negative 141.0

neutral 217.0

positive 454.0

Surprisingly there is a substantial correlation showing that as the email length increases the emails get more positive. This shows that if they want to deliver more positive messages, they will be putting more effort into the email. Also, this was apparent in reject and non_reject as the average sentiment was fairly close together 0.77 for reject and 0.71 for non-reject showing also, they might want to end things amicably with the receiver.

Token analysis.

Method

Pre-processing of the tokens was done with removing stop words and non-alphabetic characters. Then, exploded the list-like elements to rows then counted how often the item appeared. The item refers to single word, biagram and triagram to find a wide range of words and combinations and seeing trends there. This was also done on exclusively the first 5 and last 5 tokens of each email as well. They are split into reject and non_reject results as seen in appendices.

Results

Word frequency

Both reject and non-rejection emails have a lot of appearances of the word “software” and “engineer” which shows that even his non-rejection emails have a lot to do with topics surrounding jobs and software engineering due words like resume and job appearing 26 times. The rejection emails seem more similar with 79 references to the word interest in rejection emails with second and third one being closely situated (68 – thank and 58 -position). “Unfortunately,” being used 53 times in rejection emails show that they want to formally cut off ties.

Initial tokens show that non-rejection formals are less formal since they use “hi” 9 times while it isn’t even seen in rejection emails. Non-rejection emails want more engagement which is why they use words like please 28 times and get 19 times which show an action is being asked.

Biagram and Trigram frequency

These were used to reaffirm the conclusions made earlier with word frequency of the tokens.

The emails are very focused about software engineering as it was referenced 32 times in reject emails and 24 times in non-reject. Seth is from US which can be seen as US was mentioned 5 times.

Also demanding interest was a correct conclusion as it was mentioned lots of times in different ways such as let know - 12 times and please feel free - 3 times. would consider joining - 3 times.

The triagrams at the start are very similar in rejection emails as well as other parts which shows how much similar rejection emails are. The main conclusion drawn from this section is that rejection emails are very similar to each other due to having lots of common phrases and words.

Also, the whole dataset is very focused and biased towards a software engineer's kind of emails as there were lots of mentions to software engineers and job looking and jobs in general.

Evaluation

The methods used gave us a lot of insight about the dataset and can serve as a good basis for future work. All the results seem obtained are valid for the dataset. All the methods were suited for the purposes of analysis. Stemmers were not used due to how aggressive they get rid of stems which in the field of computer science related terminology shouldn't be done in my opinion since it leads to words relating to computer science losing its meaning fully.

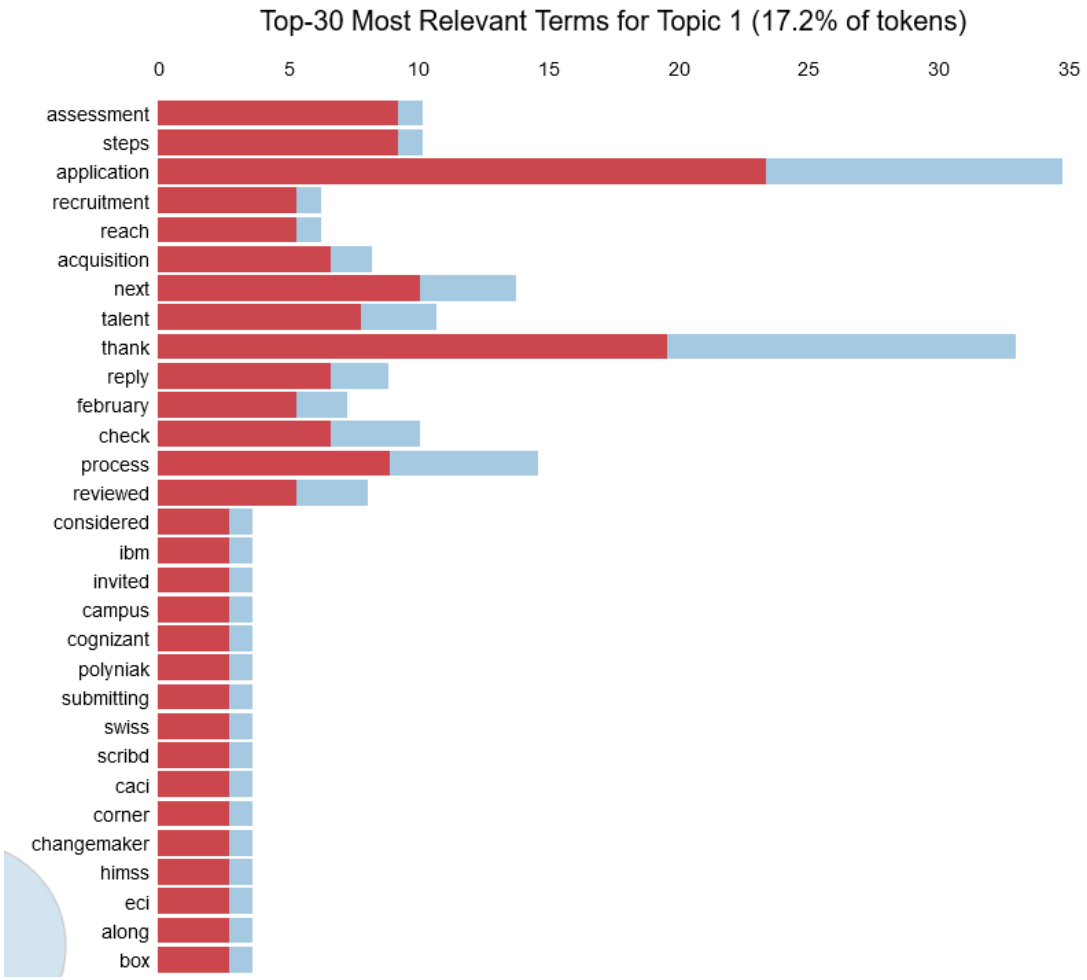
Given we hit some obstacles with the data, a bigger question to think is how applicable and useful is the dataset in general?

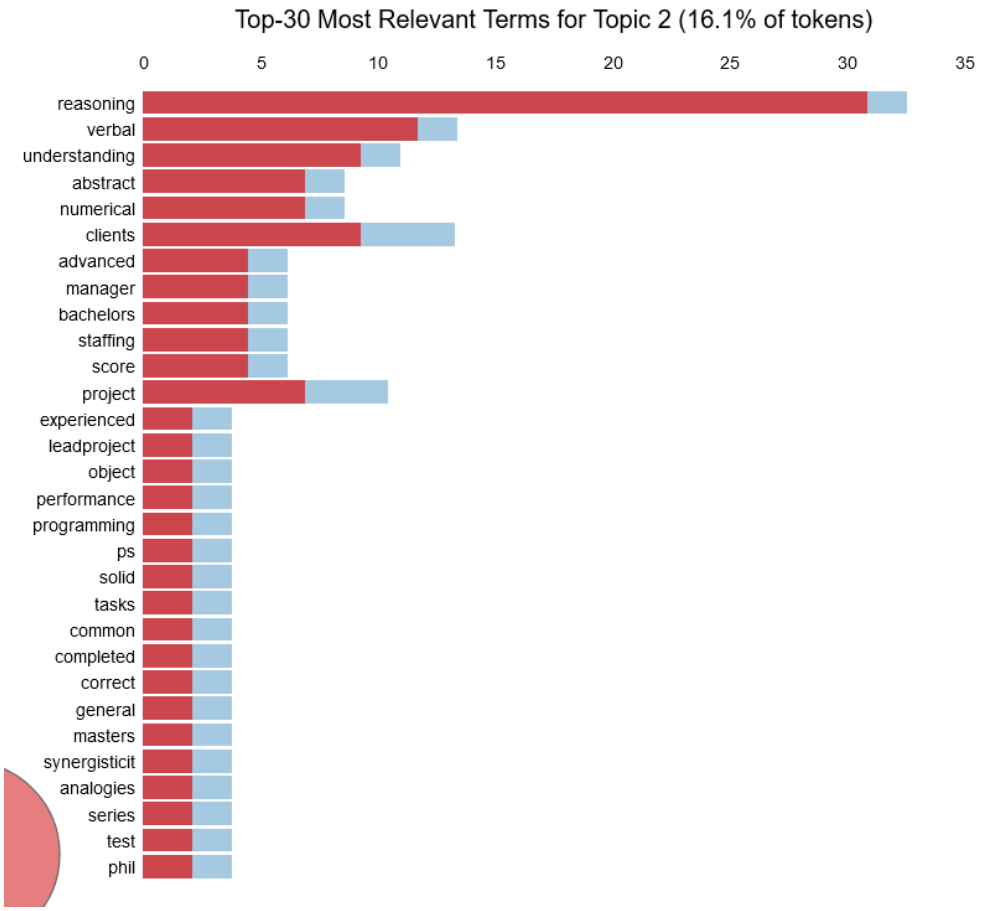
Limitations of the dataset

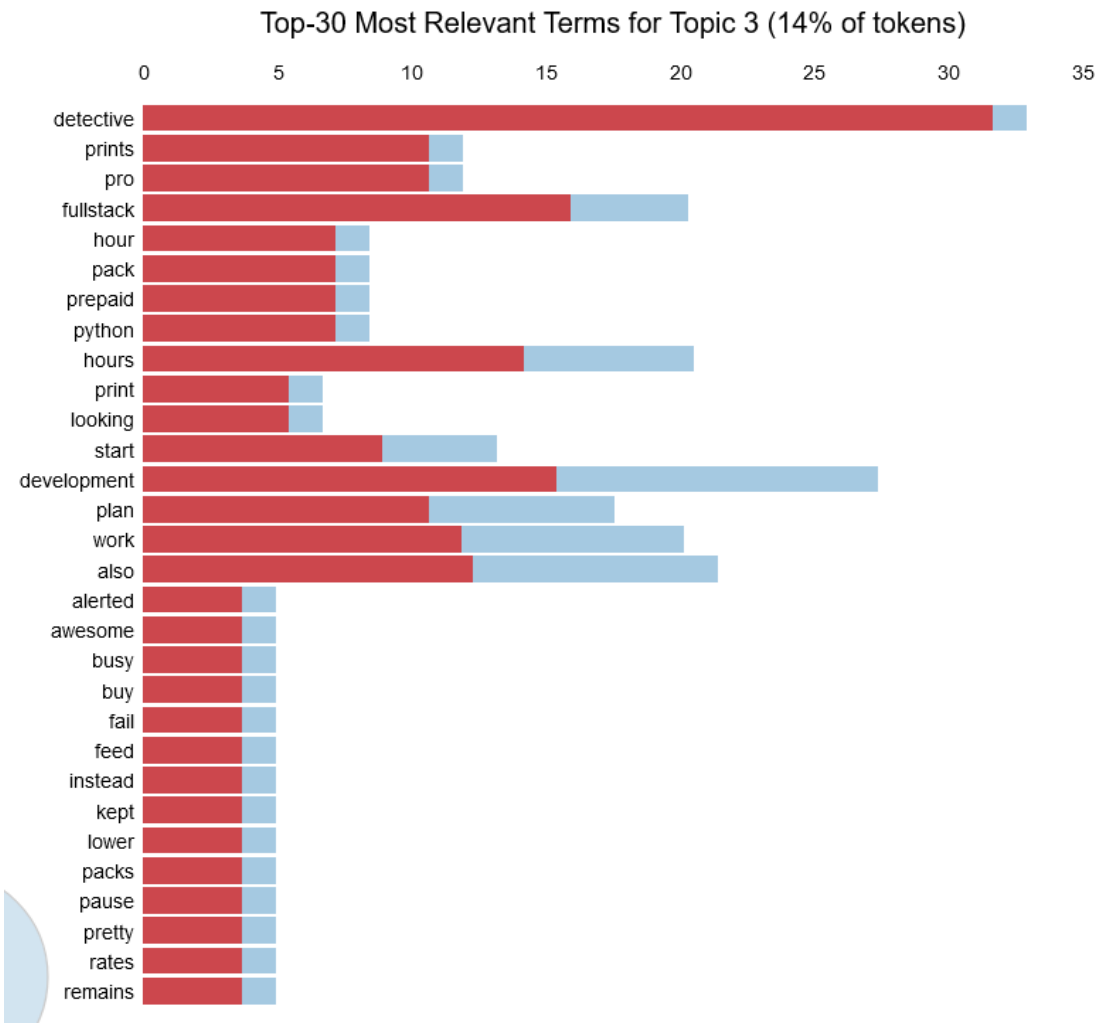
One of the main limitations of the dataset is that it is extremely small, this will limit the usage of the dataset a lot since the patterns found here don't have lots of data to build from and can lead to overfit data. The size of the dataset can be seen limiting the information gathered in this report highlighting the size is a problem. Another reason for overfitting in models created for this dataset is due to how biased this information is, models created, and analysis will be more useful for software engineer jobs since other jobs can have completely different formats for rejection and Seth's non-rejection emails are also mainly about jobs due to his main focus around the time the emails were taken from being about jobs in the Computer Science field. One additional data for analysis that would be very useful is temporal data as patterns can emerge there and more in-depth analysis can be done through that.

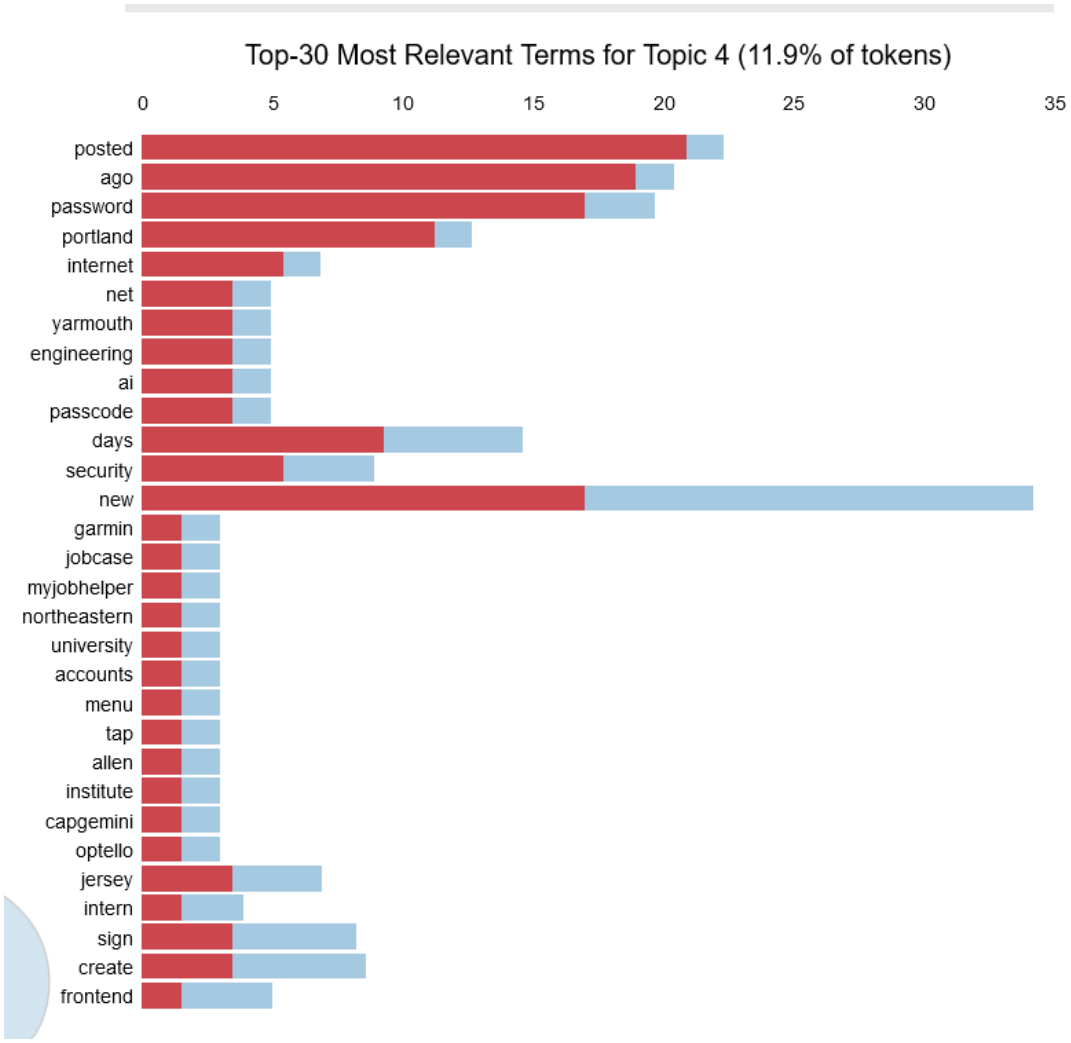
Appendices

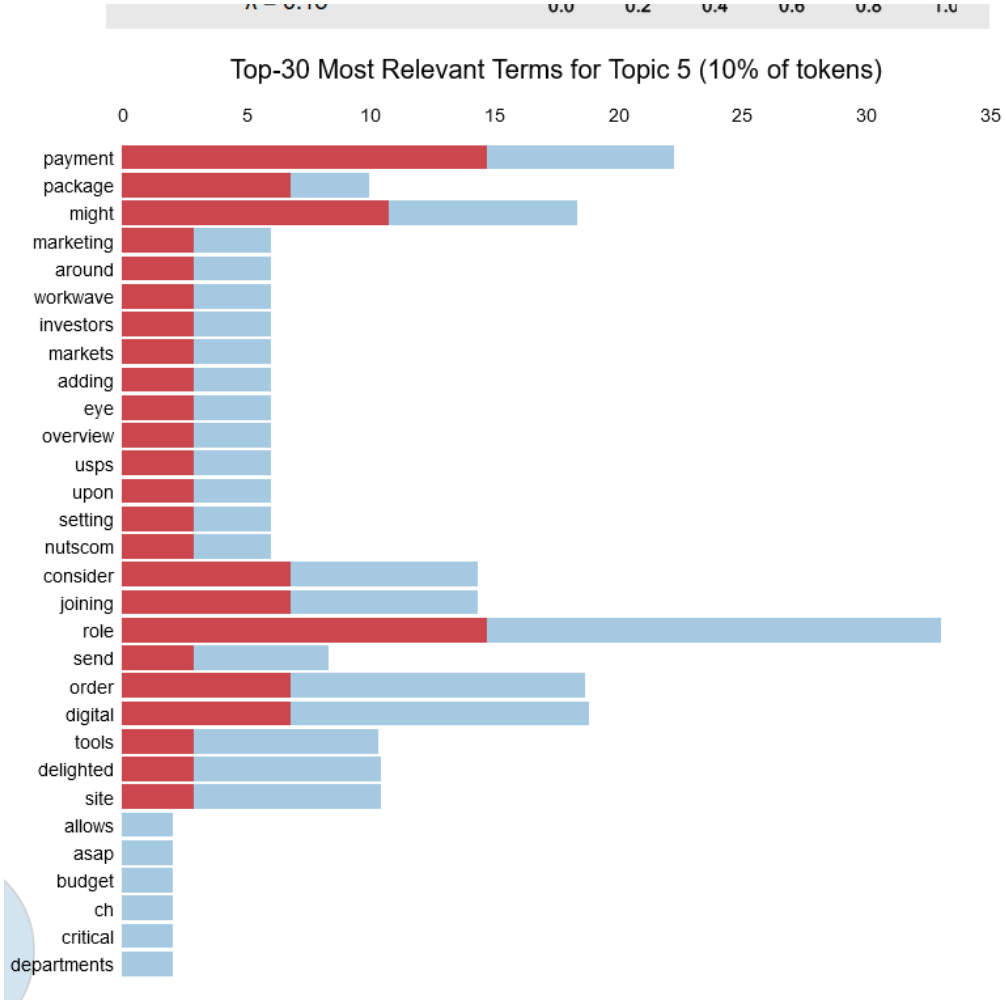
LDA – reject in detail

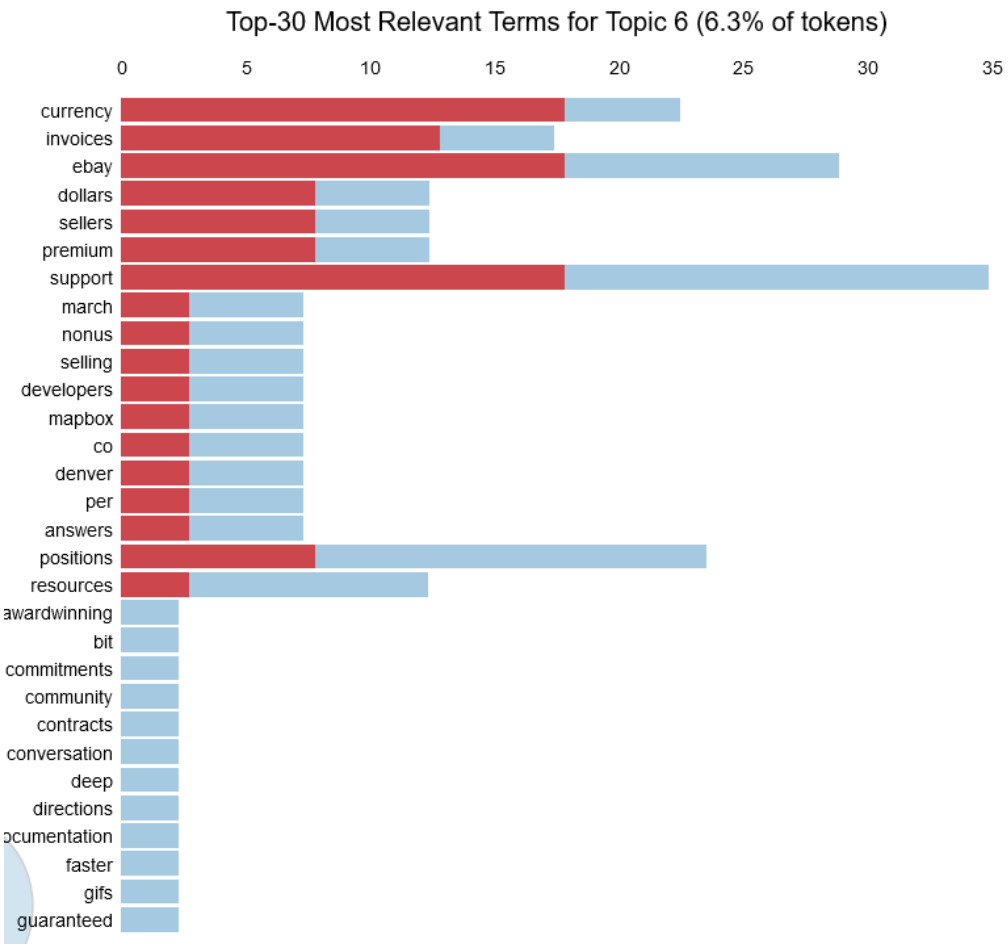


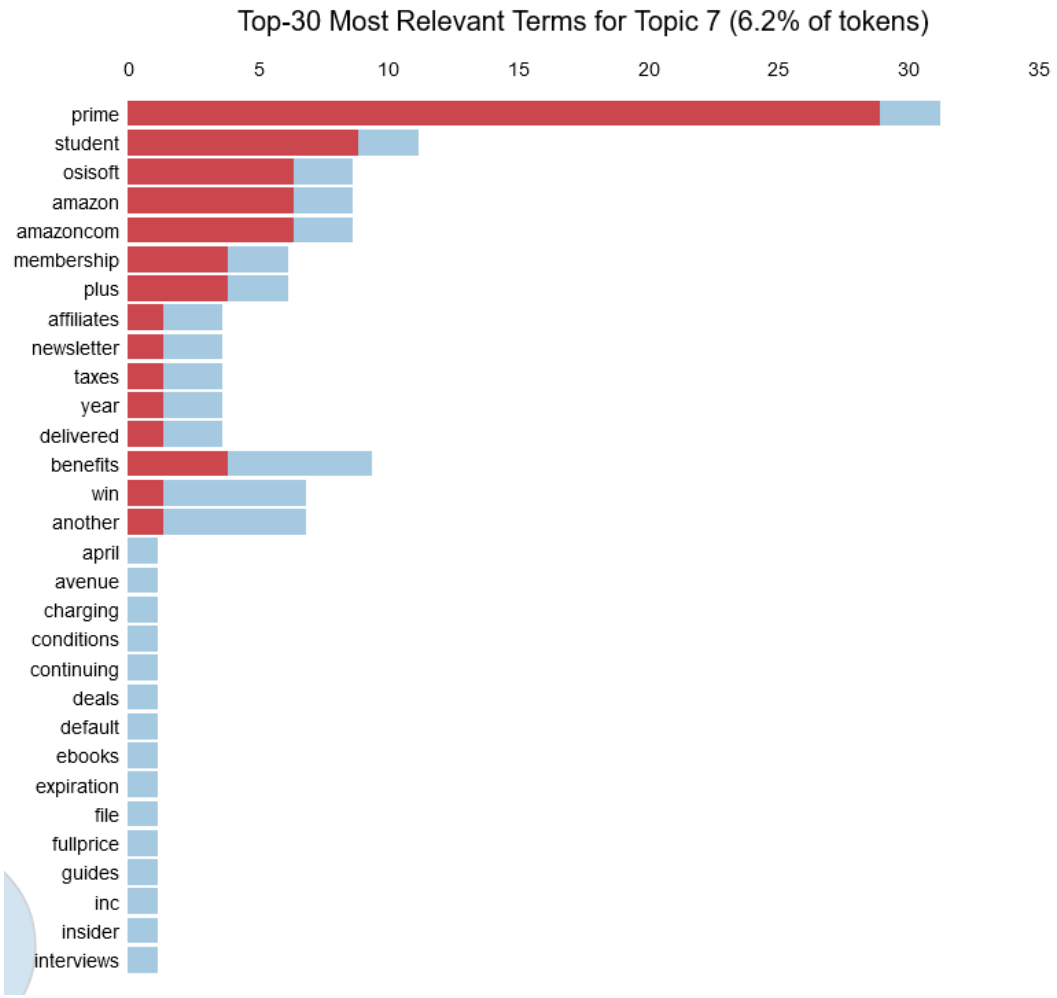

Top-30 Most Relevant Terms for Topic 2 (16.1% of tokens)
0
5
10
15
20
25
30
35
reasoning
verbal
understanding
abstract
numerical
clients
advanced
manager
bachelors
staffing
score
project
experienced
leadproject
object
performance
programming
ps
solid
tasks
common
completed
correct
general
masters
synergisticit
analogies
series
test
phil


Top-30 Most Relevant Terms for Topic 3 (14% of tokens)
0
5
10
15
20
25
30
35
detective
prints
pro
fullstack
hour
pack
prepaid
python
hours
print
looking
start
development
plan
work
also
alerted
awesome
busy
buy
fail
feed
instead
kept
lower
packs
pause
pretty
rates
remains


Top-30 Most Relevant Terms for Topic 4 (11.9% of tokens)
0
5
10
15
20
25
30
35
posted
ago
password
portland
internet
net
yarmouth
engineering
ai
passcode
days
security
new
garmin
jobcase
myjobhelper
northeastern
university
accounts
menu
tap
allen
institute
capgemini
optello
jersey
intern
sign
create
frontend


0.0
0.2
0.4
0.6
0.8
1.0
Top-30 Most Relevant Terms for Topic 5 (10% of tokens)
0
5
10
15
20
25
30
35
payment
package
might
marketing
around
workwave
investors
markets
adding
eye
overview
usps
upon
setting
nutscom
consider
joining
role
send
order
digital
tools
delighted
site
allows
asap
budget
ch
critical
departments


Top-30 Most Relevant Terms for Topic 6 (6.3% of tokens)
0
5
10
15
20
25
30
35
currency
invoices
ebay
dollars
sellers
premium
support
march
nonus
selling
developers
mapbox
co
denver
per
answers
positions
resources
awardwinning
bit
commitments
community
contracts
conversation
deep
directions
ocumentation
faster
gifs
guaranteed



NMF

Non Reject

Topic 0:companies services tata tcs north consultancy description america acquisition talent

Topic 1:free thanks help feel assistance visit contact reviewed payment com

Topic 2:100 ma positions jersey new states united engineer developer software

Topic 3:start pre hour paid pack plan prints pro hours detective

Topic 4:2021 development delivered package delivery seller rate seth experience hi

Topic 5:verification clicking thank email link registering site career verify account

Topic 6:like quickly ziprecruiter profile sincerely login update job ve resume

Topic 7:questions recruitment hello wanted seth reach steps recent thank application

Topic 8:chance winner 23 est fates remember shining win 2021 february

Topic 9:touch engineer received delighted joining consider position role team application

Topic 10:response resources started code month search step plan premium support

Topic 11:location new ebay used https device accounts change sign password

Reject

Topic 0:

time position software application forward regards appreciate united states moving

Topic 1:experience visit engineer software review com technologies career apply opportunities

Topic 2:time position application developer applying selected hiring process moved step

Topic 3:received applicants qualified particular opening ahead candidacy feel proceed decided

Topic 4:decision motors postings developer new applying job entry level gm

Topic 5:truly community journey family automotive pique phase brands haven cox

Topic 6:giving consideration recruiters http accept inquiry jobs companies employment tjx

Topic 7:continue eye careers site career fit like candidates know team

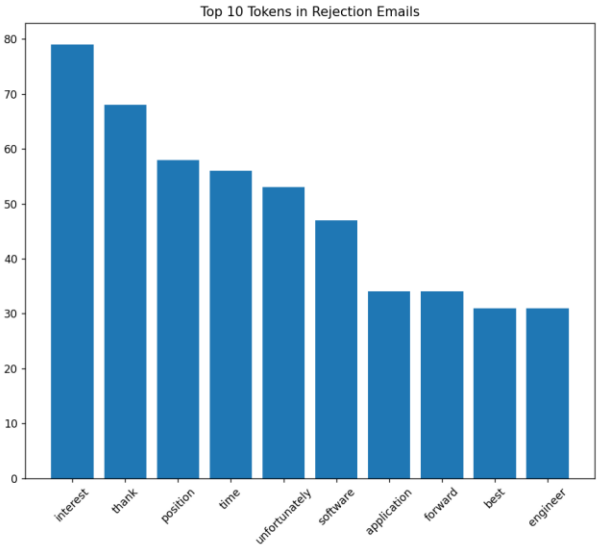
Topic 8:regards engineer candidates forward united states did massachusetts select solutions

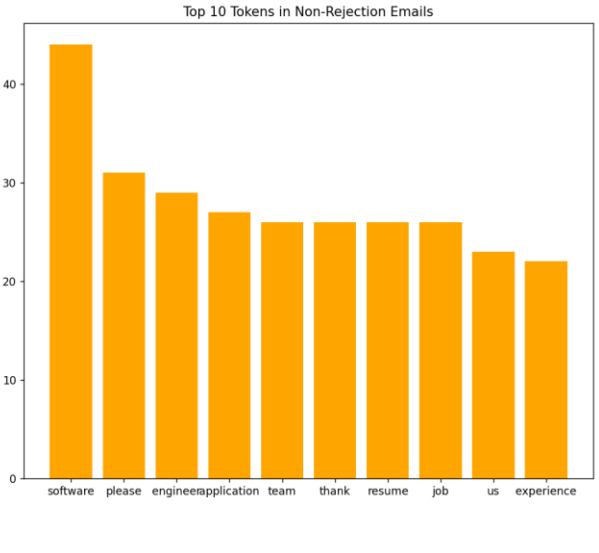
Topic 9:unfortunately stack software time application forward developer technology decided junior

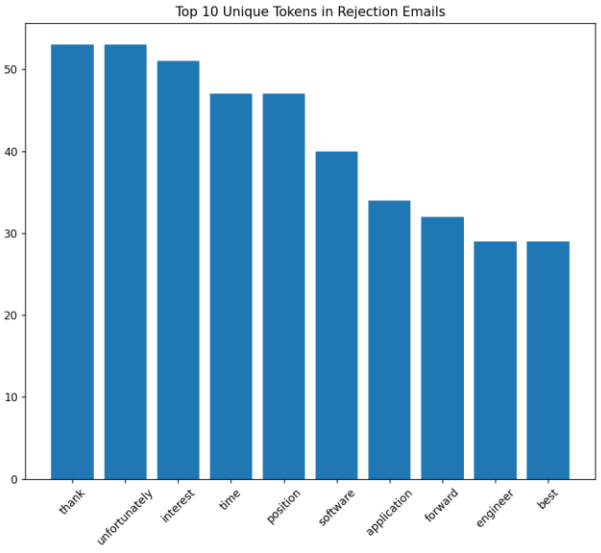
Topic 10:search matches hesitate change sincerely job opens opening employment contact

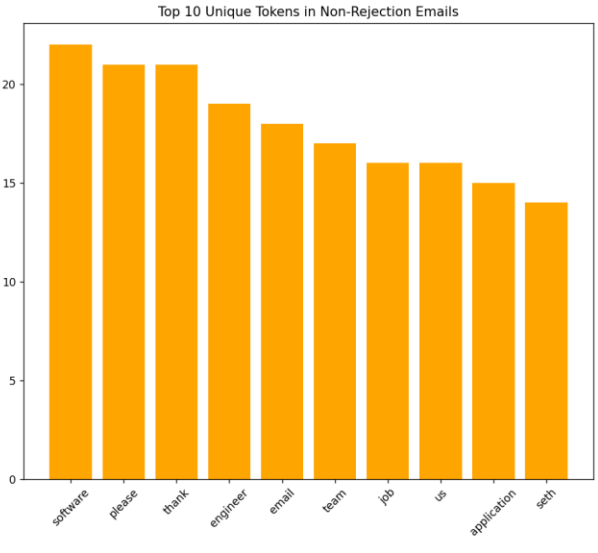
Topic 11:wish touch unable file role seth best reach hi future

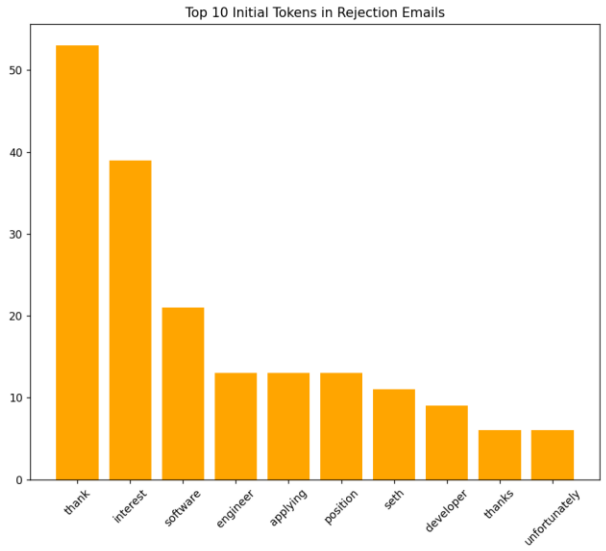
Tokens

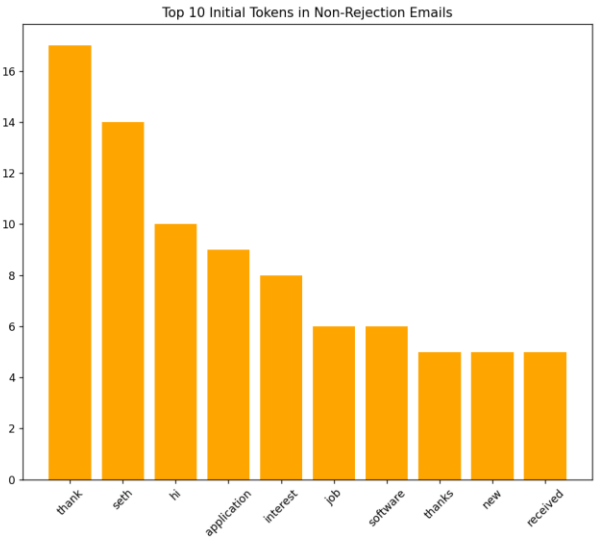

Top 10 Tokens in Rejection Emails
interest
thank
position
time
unfortunately
software
application
forward
best
engineer

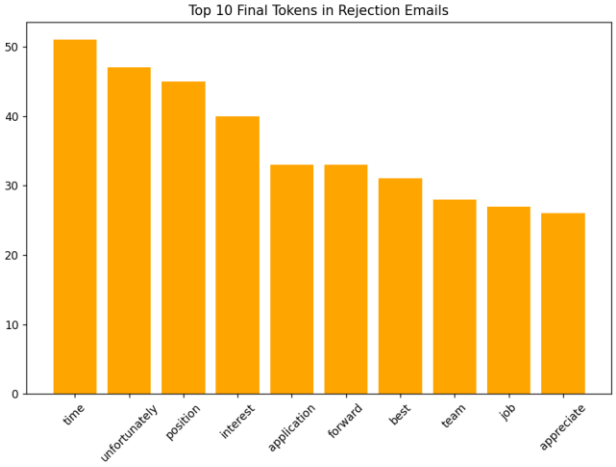

Top 10 Tokens in Non-Rejection Emails
software
please
engineer
application
team
thank
resume
job
us
experience

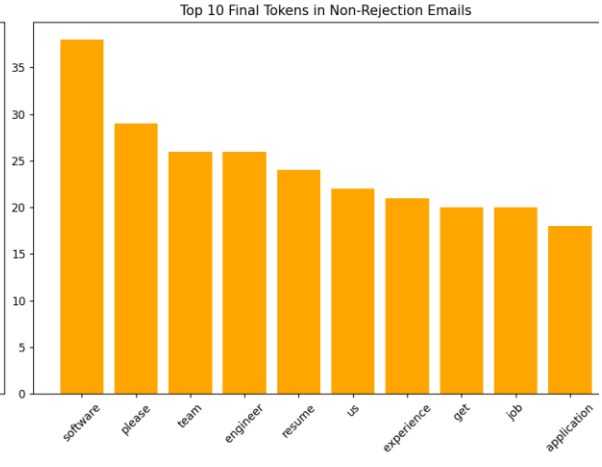

Top 10 Unique Tokens in Rejection Emails
thank
unfortunately
interest
time
position
software
application
forward
engineer
best


Top 10 Unique Tokens in Non-Rejection Emails
software
please
thank
engineer
email
team
job
us
application
seth

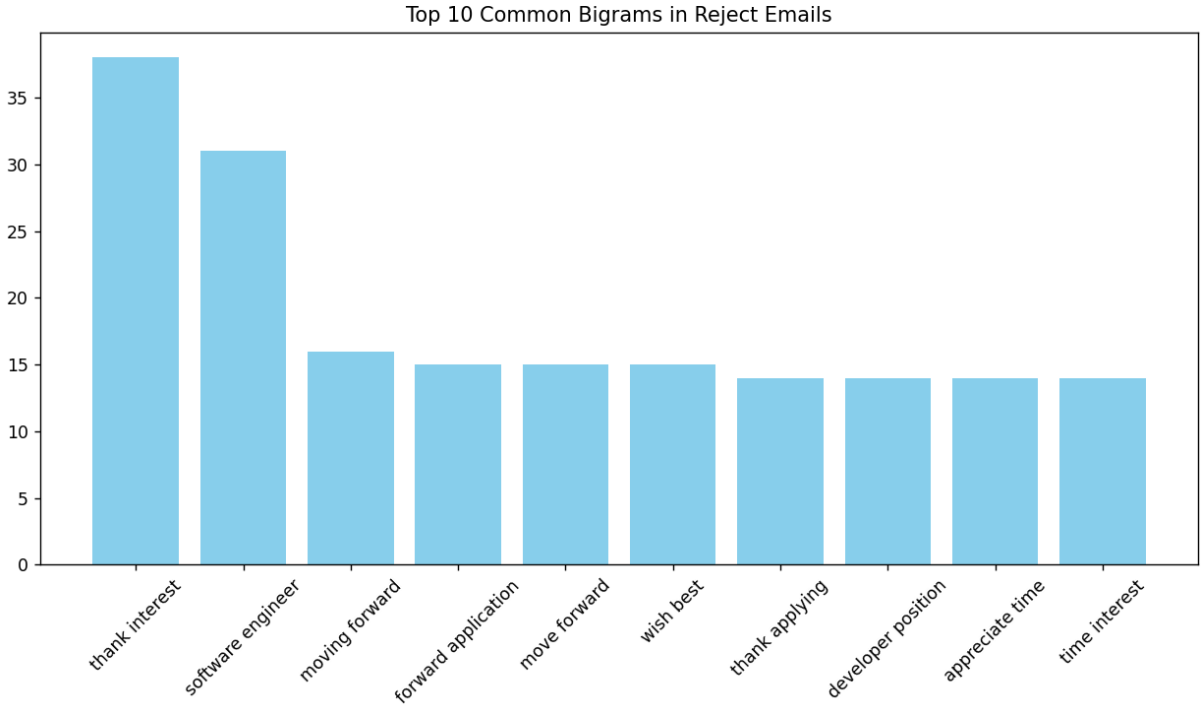

Top 10 Initial Tokens in Rejection Emails
thank
interest
software
engineer
applying
position
seth
developer
thanks
unfortunately

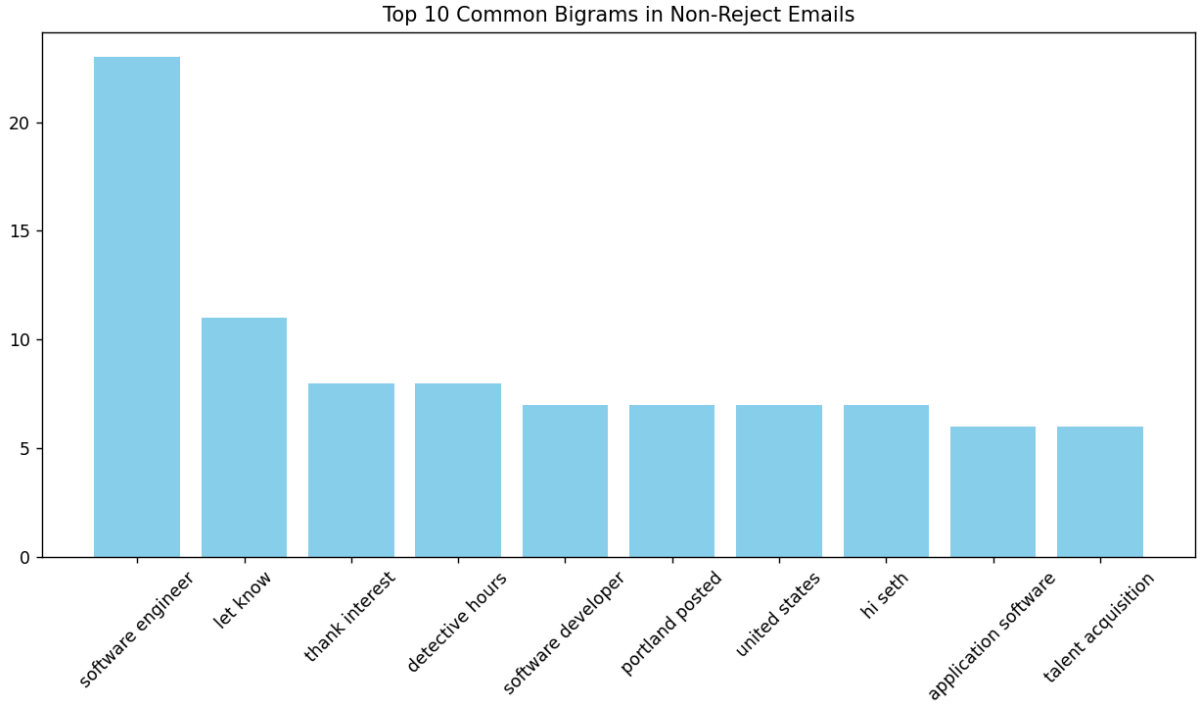

Top 10 Initial Tokens in Non-Rejection Emails
thank
seth
hi
application
interest
job
software
thanks
new
received

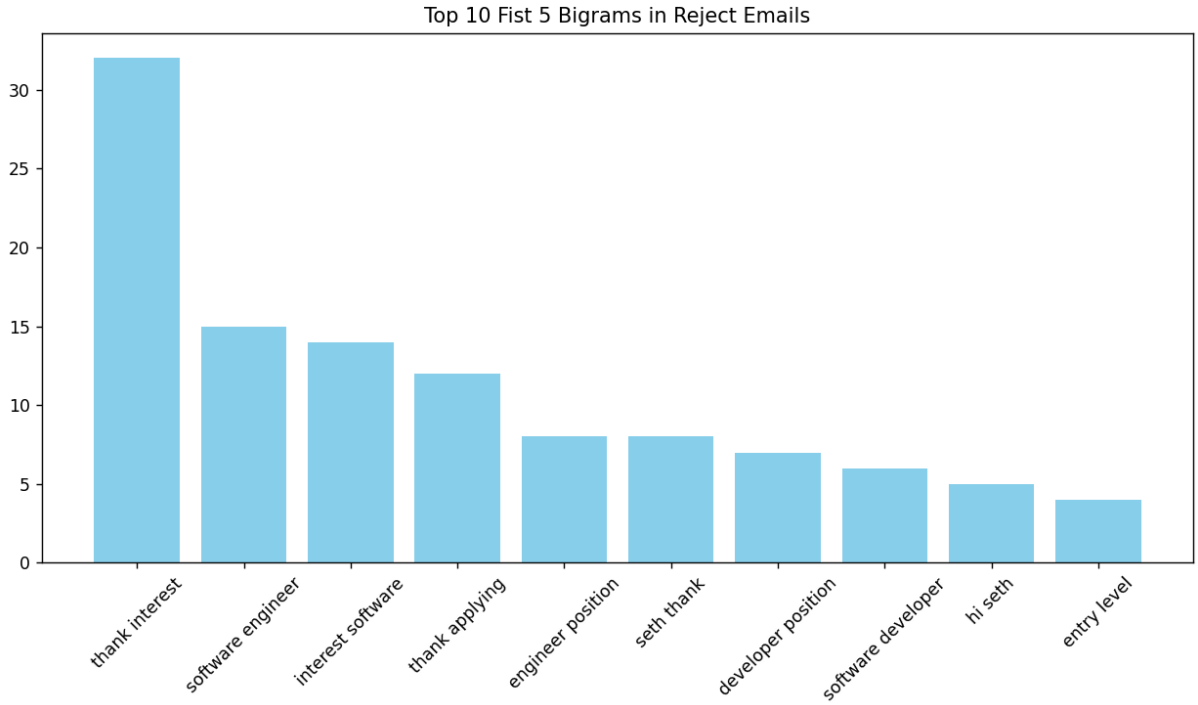

Top 10 Final Tokens in Rejection Emails
time
unfortunately
position
interest
application
forward
best
team
job
appreciate

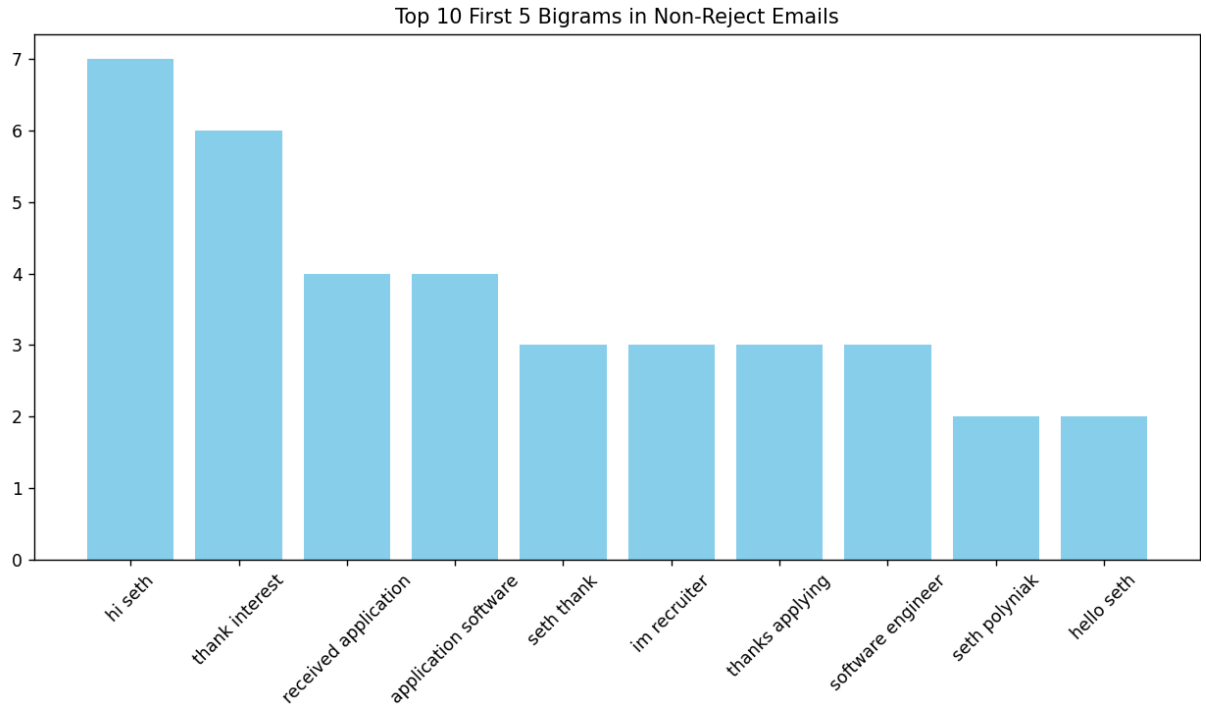

Top 10 Final Tokens in Non-Rejection Emails
software
please
team
engineer
resume
us
experience
get
job
application

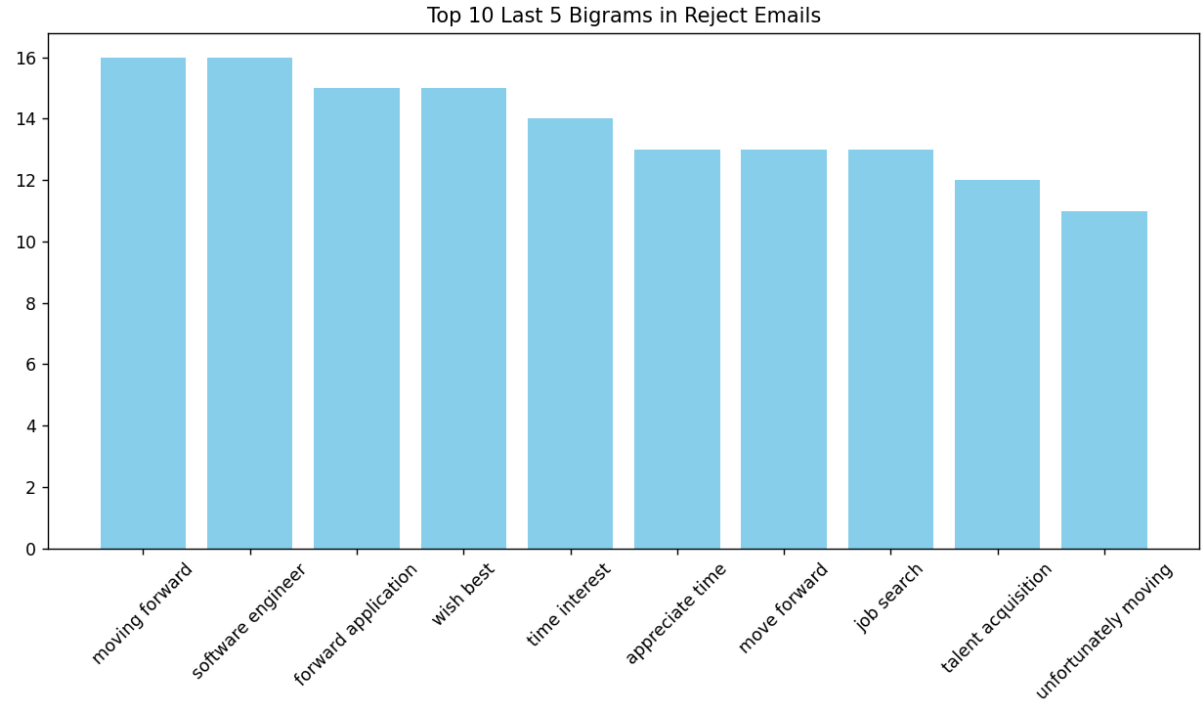
Bigrams

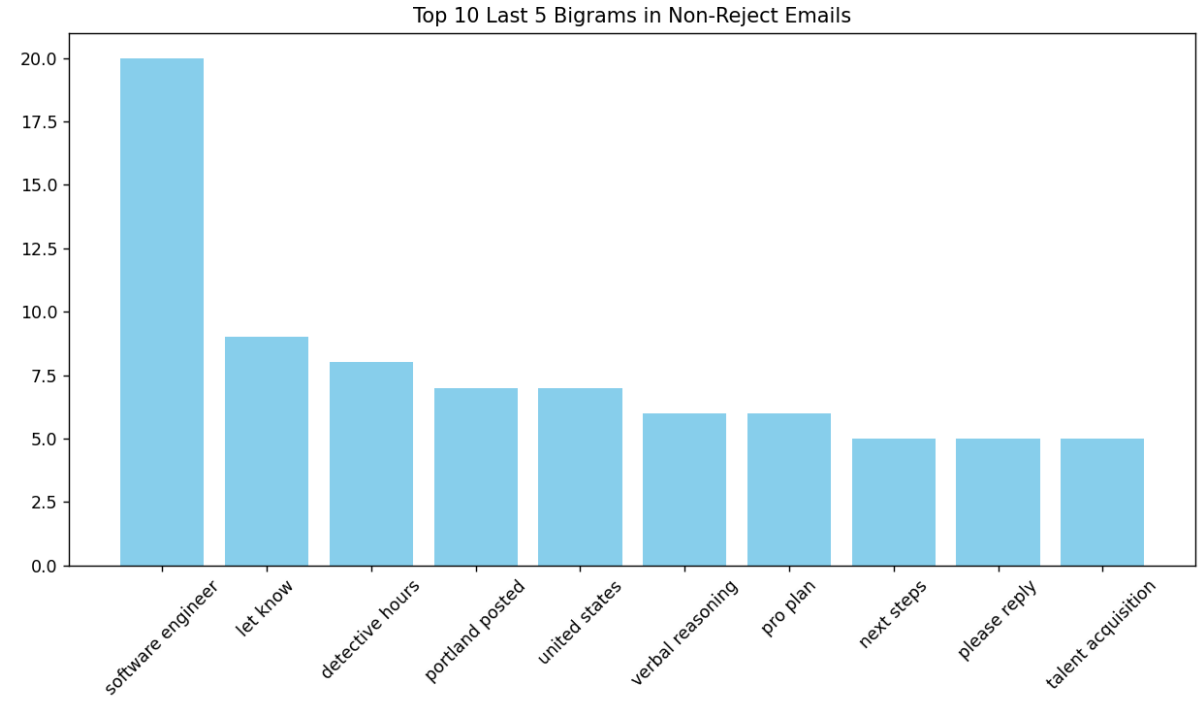

Top 10 Common Bigrams in Reject Emails
0
5
10
15
20
25
30
35
thank interest
software engineer
moving forward
forward application
move forward
wish best
thank applying
developer position
appreciate time
time interest


Top 10 Common Bigrams in Non-Reject Emails
0
5
10
15
20
software engineer
let know
thank interest
detective hours
software developer
portland posted
united states
hi seth
application software
talent acquisition

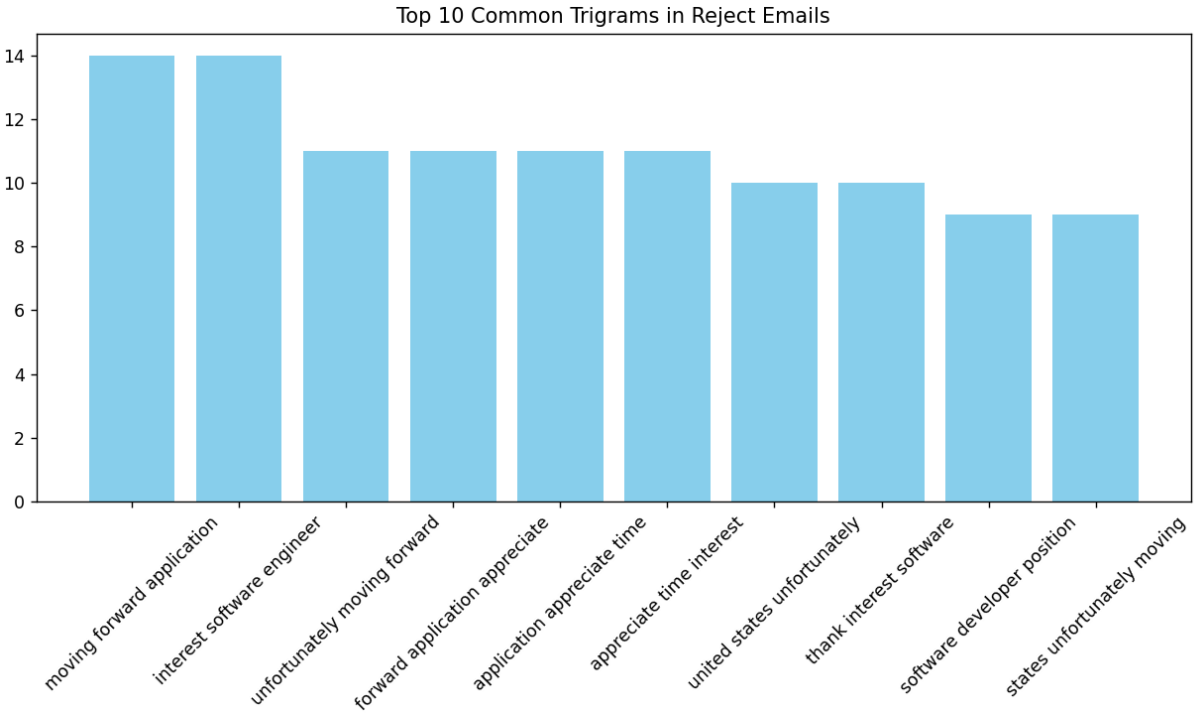

Top 10 Fist 5 Bigrams in Reject Emails
30
25
20
15
10
5
0
thank interest
software engineer
interest software
thank applying
engineer position
seth thank
developer position
software developer
hi seth
entry level

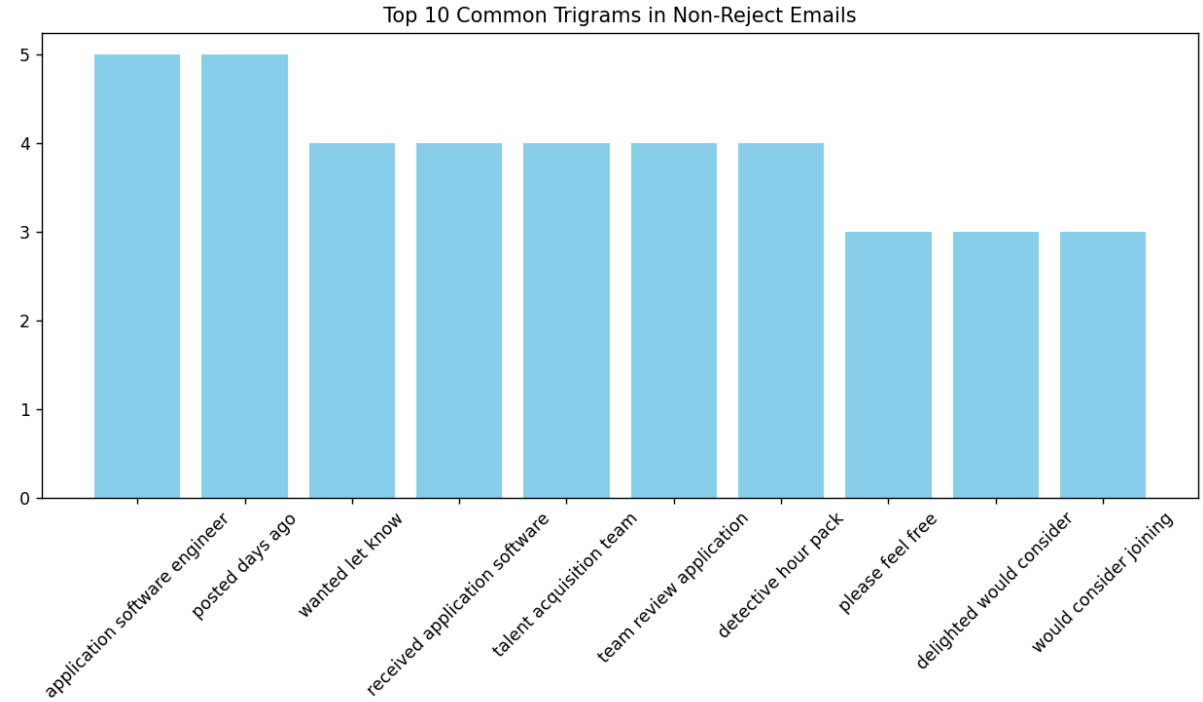

Top 10 First 5 Bigrams in Non-Reject Emails
7
6
5
4
3
2
1
0
hi seth
thank interest
received application
application software
seth thank
im recruiter
thanks applying
software engineer
seth polyniak
hello seth

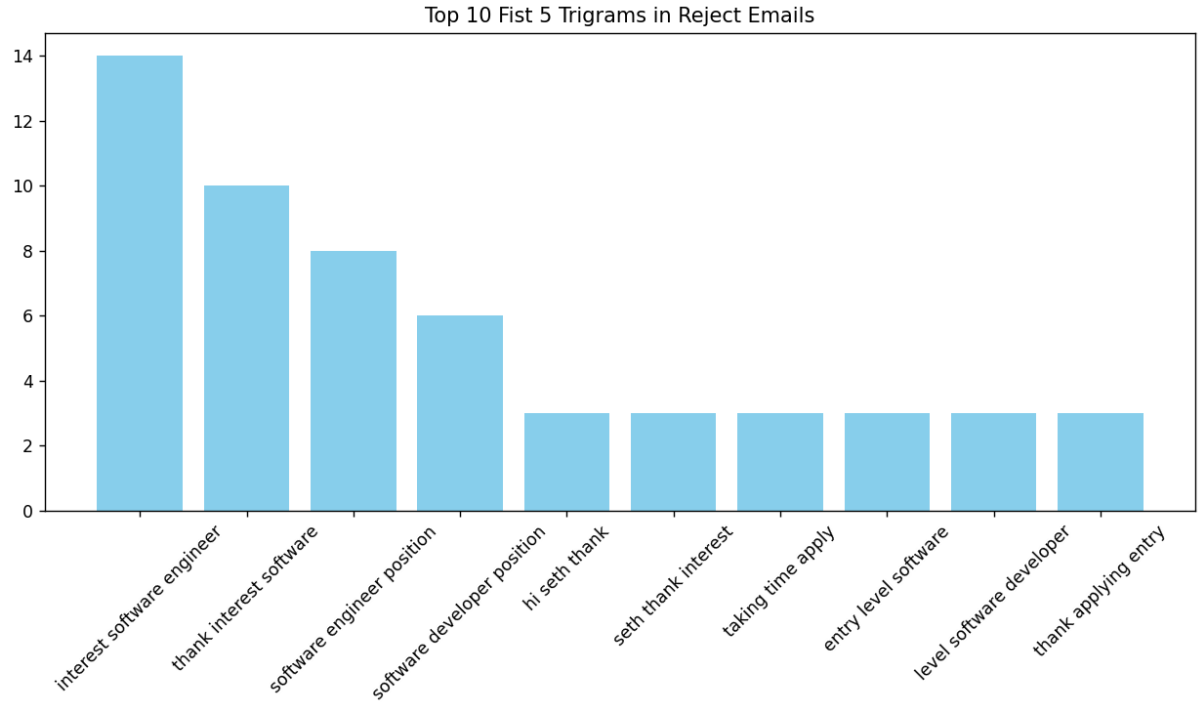

Top 10 Last 5 Bigrams in Reject Emails
16
14
12
10
8
6
4
2
0
moving forward
software engineer
forward application
wish best
time interest
appreciate time
move forward
job search
talent acquisition
unfortunately moving

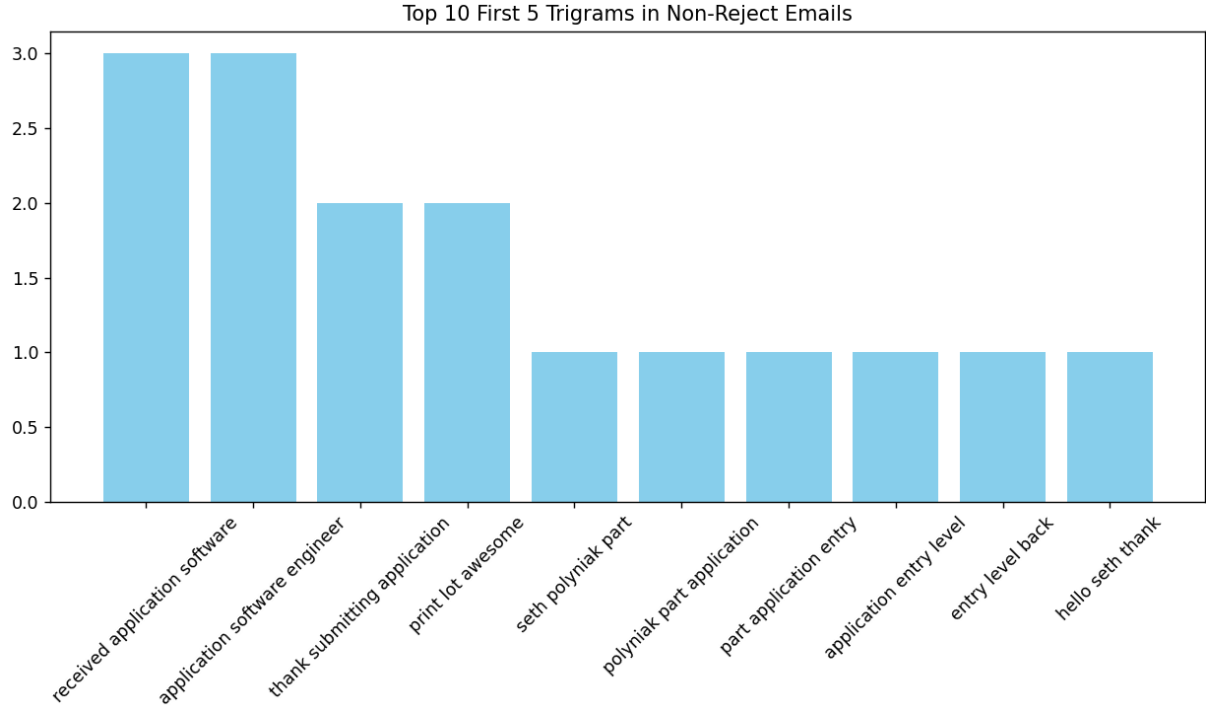

Top 10 Last 5 Bigrams in Non-Reject Emails
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
software engineer
let know
detective hours
portland posted
united states
verbal reasoning
pro plan
next steps
please reply
talent acquisition

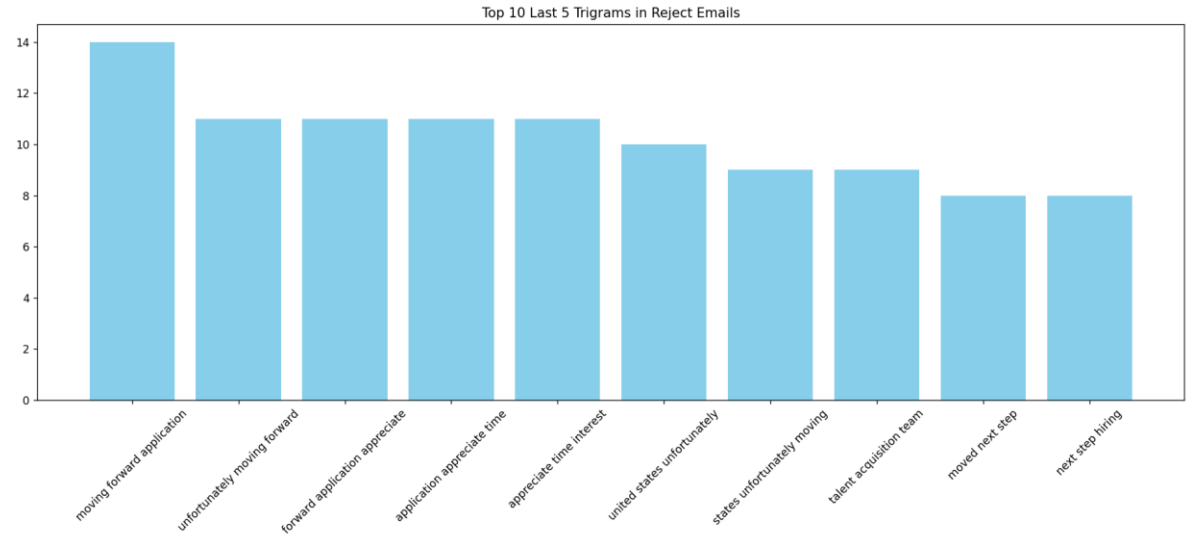
Trigrams

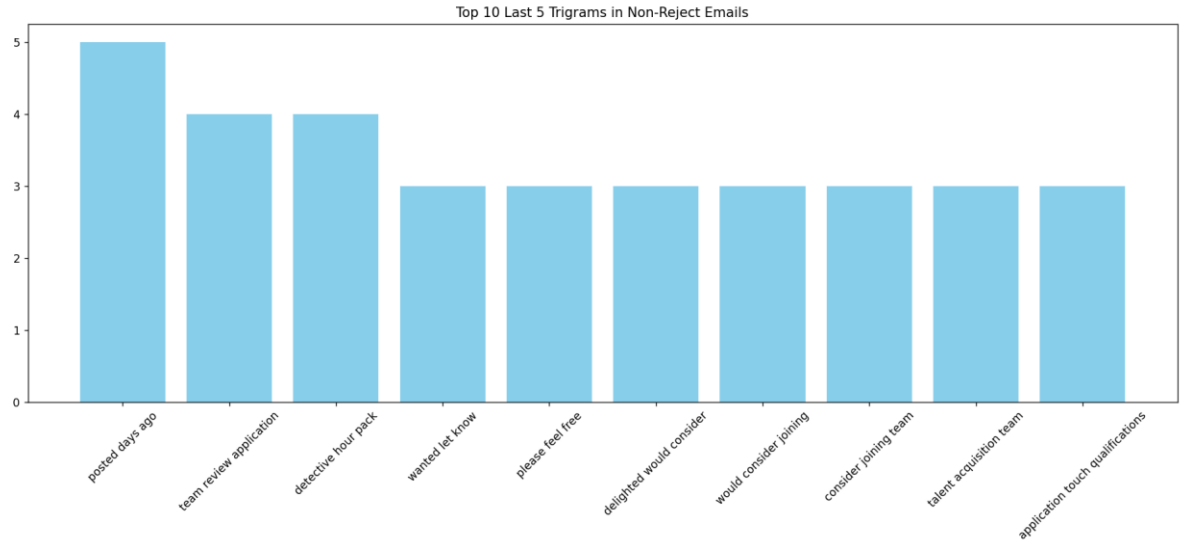

Top 10 Common Trigrams in Reject Emails
0
2
4
6
8
10
12
14
moving forward application
interest software engineer
unfortunately moving forward
forward application appreciate
application appreciate time
appreciate time interest
united states unfortunately
thank interest software
software developer position
states unfortunately moving


Top 10 Common Trigrams in Non-Reject Emails
0
1
2
3
4
5
application software engineer
posted days ago
wanted let know
received application software
talent acquisition team
team review application
detective hour pack
please feel free
delighted would consider
would consider joining


Top 10 Fist 5 Trigrams in Reject Emails
14
12
10
8
6
4
2
0
interest software engineer
thank interest software
software engineer position
software developer position
hi seth thank
seth thank interest
taking time apply
entry level software
level software developer
thank applying entry


Top 10 First 5 Trigrams in Non-Reject Emails
3.0
2.5
2.0
1.5
1.0
0.5
0.0
received application software
application software engineer
thank submitting application
print lot awesome
seth polyniak part
polyniak part application
part application entry
application entry level
entry level back
hello seth thank





Code

NMF

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import NMF

data = pd.read_csv('Rejection Data - Sheet1.csv')

non_reject_emails = data[data['Status'] == 'reject']['Email']

tfidf_vectorizer = TfidfVectorizer(max_df=0.95, min_df=2,
stop_words='english')
```

```
tfidf = tfidf_vectorizer.fit_transform(non_reject_emails)

nmf_model = NMF(n_components=12, random_state=42)
W = nmf_model.fit_transform(tfidf)
H = nmf_model.components_

feature_names = tfidf_vectorizer.get_feature_names_out()

for i, topic in enumerate(H):
    print(f"Topic {i}:")
    print(" ".join([feature_names[i] for i in topic.argsort()[-10:]]))
    print("\n")
```

LDA

```
import pandas as pd
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import re
from gensim import corpora, models
import pyLDAvis.gensim_models as gensimvis
import pyLDAvis


data = pd.read_csv('Rejection Data - Sheet1.csv')

non_reject_emails = data[data['Status'] == 'not_reject']

def preprocess(text):
    text = text.lower()
    text = re.sub(r'[^a-z\s]', '', text)
    tokens = word_tokenize(text)
    stop_words = set(stopwords.words('english'))
    tokens = [word for word in tokens if word not in stop_words]
    return tokens

non_reject_emails['processed_emails'] =
non_reject_emails['Email'].apply(preprocess)
texts = non_reject_emails['processed_emails'].tolist()

dictionary = corpora.Dictionary(texts)
corpus = [dictionary.doc2bow(text) for text in texts]

lda_model = models.LdaModel(corpus, num_topics=12, id2word=dictionary,
passes=1500)
```

```
topics = lda_model.print_topics(num_words=10)
for topic in topics:
    print(topic)


vis = gensimvis.prepare(lda_model, corpus, dictionary)
pyLDAvis.save_html(vis, 'lda_visualization.html')
```

Sentiment analysis

```
import nltk
import pandas as pd
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import matplotlib.pyplot as plt

df = pd.read_csv('Rejection Data - Sheet1.csv')
sid = SentimentIntensityAnalyzer()

def SentimentScore(text):
    return sid.polarity_scores(text)['compound']

def SentimentCat(score):
    if score >= 0.05:
        return 'positive'
    elif score <= -0.05:
        return 'negative'
    else:
        return 'neutral'

df['compoundScore'] = df['Email'].apply(SentimentScore)
df['sentiment'] = df['compoundScore'].apply(SentimentCat)
df['email_length'] = df['Email'].apply(len)

avLengOfEmailbySentiment = df.groupby('sentiment')['email_length'].median()

print("Average email length by sentiment:")
print(avLengOfEmailbySentiment)
print(df[['Email', 'email_length', 'compoundScore', 'sentiment']].head())

def RemoveOutliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lowerBound = Q1 - 1.5 * IQR
    upperBound = Q3 + 1.5 * IQR
    return df[(df[column] >= lowerBound) & (df[column] <= upperBound)]

df_clean = RemoveOutliers(df, 'email_length')
```

```
df_clean = RemoveOutliers(df_clean, 'compoundScore')

plt.scatter(df_clean['email_length'], df_clean['compoundScore'], alpha=0.8)
plt.title('Email Length against Sentiment Score')
plt.xlabel('Email Length')
plt.ylabel('Sentiment Score')
plt.grid(True)
plt.show()
```

Frequency analysis

```
import pandas as pd
from nltk import bigrams
import re
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import nltk
import matplotlib.pyplot as plt
from collections import Counter

data = pd.read_csv('Rejection Data - Sheet1.csv')
stop_words = set(stopwords.words('english'))

def preprocess(text):
    text = re.sub(r'[^a-z\s]', '', text.lower())
    tokens = [word for word in word_tokenize(text) if word not in stop_words
and len(word) > 1]
    return list(bigrams(tokens)) #can be used with [5:] or [:5] to get first 5
and last 5 tokens and also trigrams() to get trigrams

data['processed_bigrams'] = data['Email'].apply(preprocess)


def Count(df, status):
    biagramList = [bigram for sublist in df[df['Status'] ==
status]['processed_bigrams'] for bigram in sublist]
    biagramCount = Counter(biagramList)
    return biagramCount.most_common(10)

rejectemaisl = Count(data, 'reject')
nonrejectemails = Count(data, 'not_reject')

def drawGraph(numOfBiagrams, title, graph):
    bigrams, counts = zip(*numOfBiagrams)
    bigrams = [' '.join(bigram) for bigram in bigrams]
    graph.bar(bigrams, counts, color='blue')
    graph.set_title(title)
    graph.tick_params(graphis='x', rotation=45)
```

```
fig, graphs = plt.subplots(1, 2)
drawGraph(rejectemaisl, 'Top 10 Bigrams in Rejection Emails', graphs[0])
drawGraph(nonrejectemails, 'Top 10 Bigrams in Non-Rejection Emails',
graphs[1])
plt.show()
```