Project: Conversation Dialogue Identification

York University CSML1010 – Milestone 1 Group 3: Jerry Khidaroo, Paul Doucet

Instructor: Dr. Annie En-Shiun Lee

CSML1010 – Milestone 1

- Dataset: Taskmaster-1 from Google
- NLP Multi-Class Text Classification Problem
- Data Preparation
- Data Clean Up
- Exploratory Data Analysis
- **Feature Extraction & Engineering**
- **Feature Scaling & Selection**
- Modeling
- Model Evaluation & Tuning

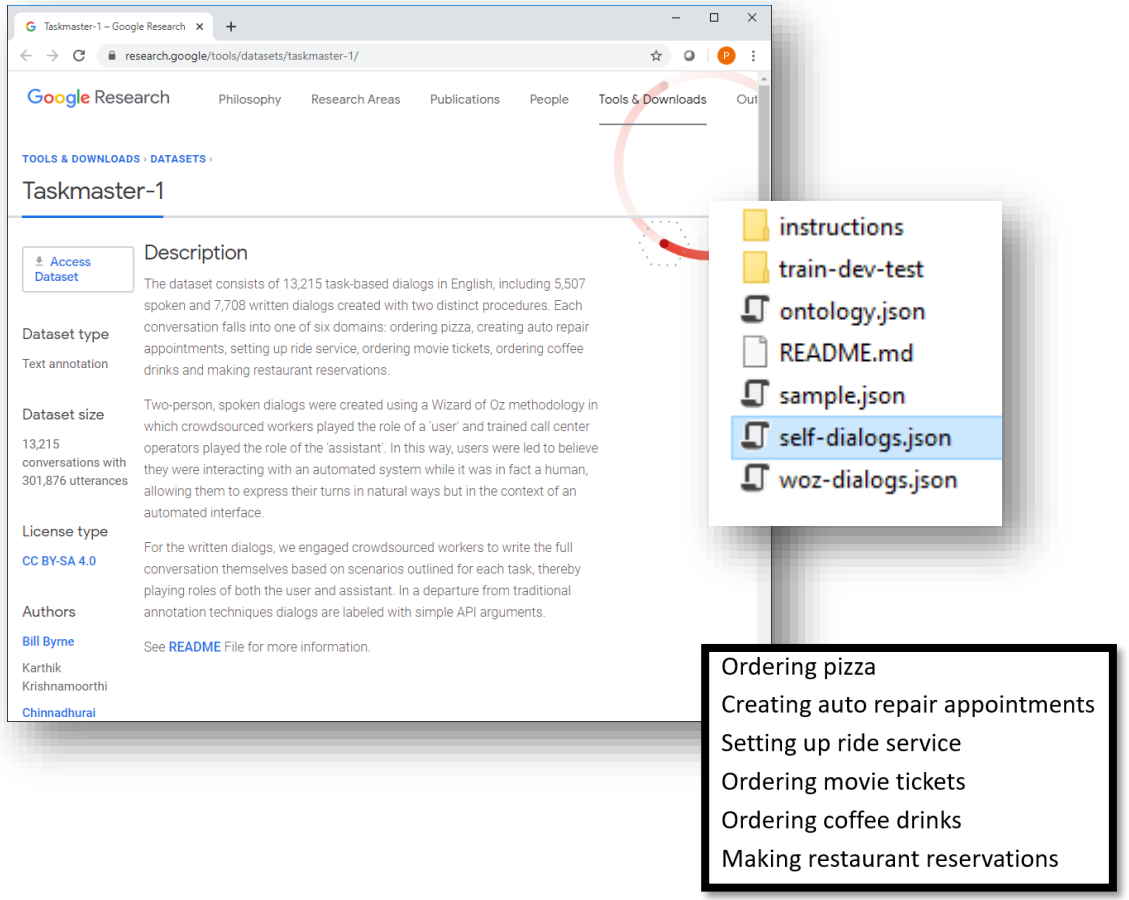
Dataset: Taskmaster-1 from Google

https://research.google/tools/datasets/taskmaster-1/

The dataset selected is the Taskmaster-1 from Google

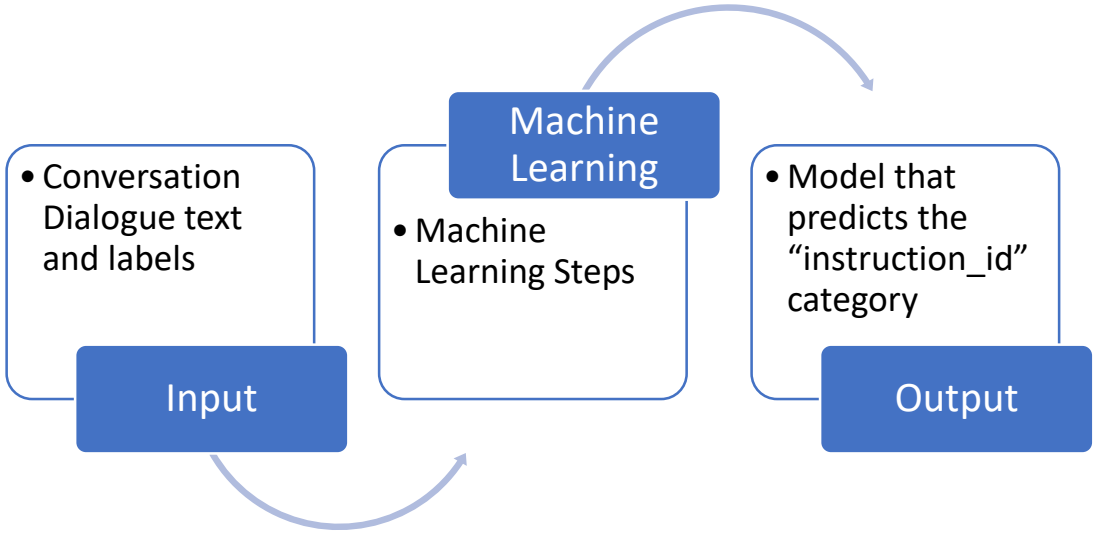
The dataset consists of task-based dialogs falling into one of six domains divided into 14 categories

We will be using the self-dialogs file which contains 7,708 conversations: **‘self-dialogs.json’**



Project: Conversation Identification

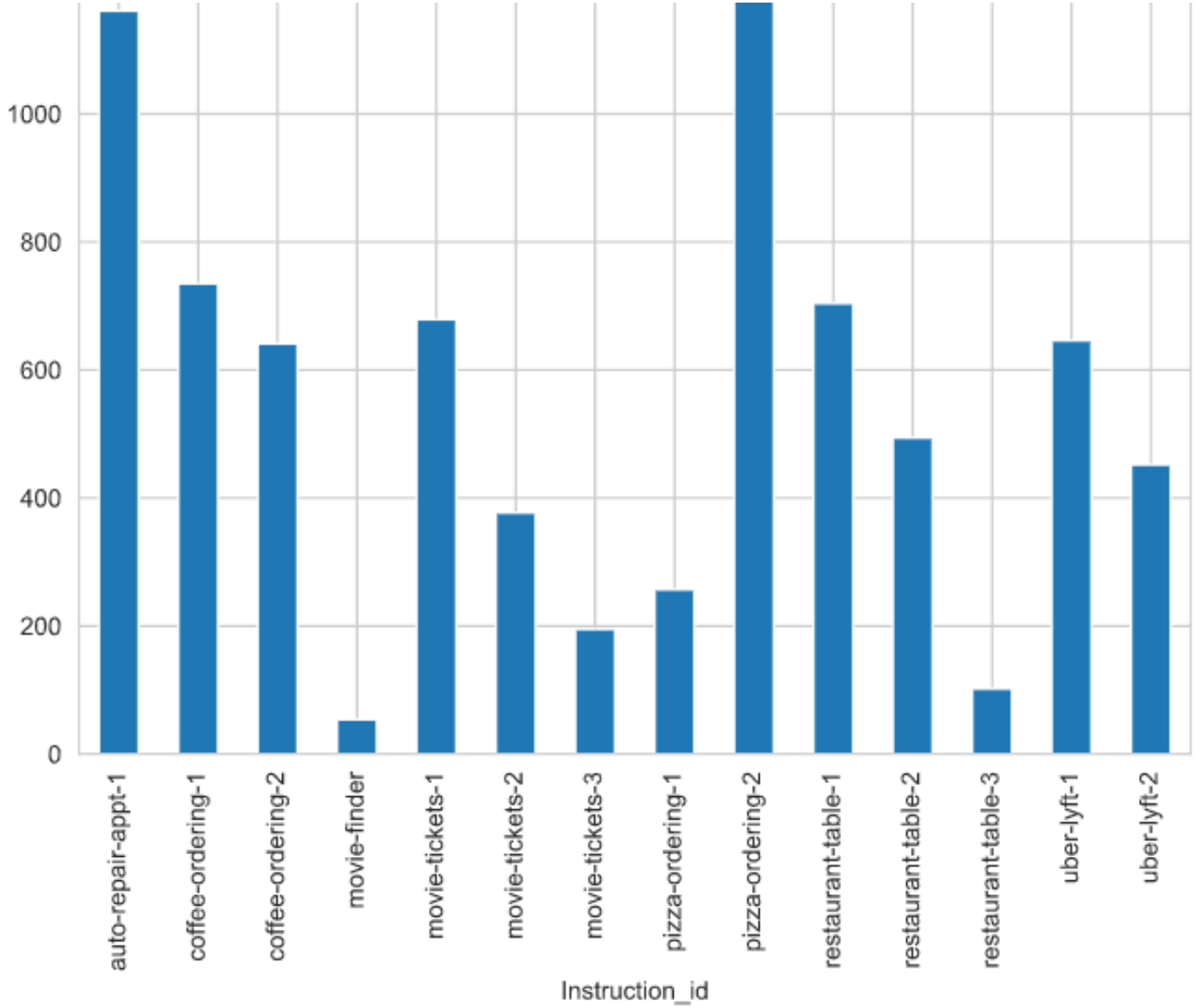
- The problem we will examine is a supervised multi-class text classification problem.
- **Goal:** *Build a model that identifies the category for dialogue conversations*
  - **Input:** conversation dialogue text and labels
  - **Output:** model that predicts the “instruction_id” category



Description of Categorical Variable (**Instruction_id**)

Instruction_id	Conversation Dialogue Description
Auto-repair-appt-1	Users will pretend they need to take their car to the mechanic, so they need to get an appointment scheduled
Coffee-ordering-1	Users will pretend they've decided to order a coffee drink from a coffee shop
Coffee-ordering-2	Users will pretend they've decided to order a coffee drink, and makes changes to the drink after the initial options have been requested
movie-finder	User is looking for a movie to see at home
movie-tickets-1	User wants to see a movie playing now
movie-tickets-2	User wants to see a movie playing now, settling for a second choice
movie-tickets-3	User wants to see one of two movies
Pizza-Ordering-1	User orders one pizza, and ask all relevant details
Pizza-Ordering-2	User orders one pizza with two toppings, and ask all relevant details
restaurant-table-1	Users will pretend they are searching for a restaurant and book a table
restaurant-table-2	Users will pretend they are searching for a restaurant and book a table, and will need to find an alternative when choice is not available
restaurant-table-3	Users will pretend they are searching for a restaurant and book a table, and will look at options at two restaurants
uber-lyft-1	Users will pretend they need to order a car for a ride inside a city
uber-lyft-2	Users will pretend they need to order a car for a ride inside a city, and looking for an alternative when choice is not available

Distribution of Categorical Variable (Instruction_id)



Balanced Sample of Dataset

Down sample to 1000 records.

```
print ((1000 * 1000)//7708)
```

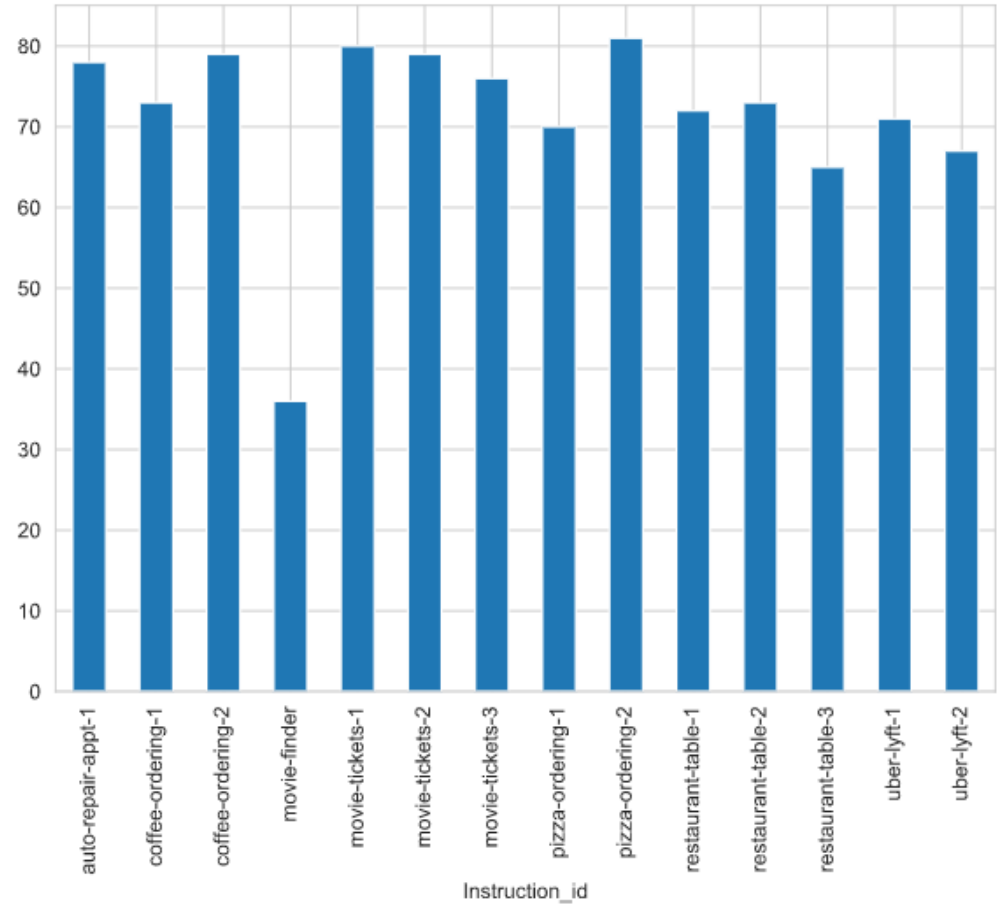
```
129
```

```
# Get 50 per instruction_id then reduce to 1000
def sampling_k_elements(group, k=130):
    if len(group) < k:
        return group
    return group.sample(k)

#Get balanced samples
corpus_df = df_all.groupby('Instruction_id').apply(sampling_k_elements).reset_index(drop=True)

#Reduce to 1000
corpus_df = corpus_df.sample(n=1000, random_state=1)
print (corpus_df.groupby('Instruction_id').size())
```

```
Instruction_id
auto-repair-appt-1    78
coffee-ordering-1     73
coffee-ordering-2     79
movie-finder          36
movie-tickets-1       80
movie-tickets-2       79
movie-tickets-3       76
pizza-ordering-1      70
pizza-ordering-2      81
restaurant-table-1    72
restaurant-table-2    73
restaurant-table-3    65
uber-lyft-1           71
uber-lyft-2           67
dtype: int64
```



Feature Extraction and Selection Summary

Vector Types	Feature Extracted	Scaling	Feature Selection Method
Count Vectors	Bag-of-words	MinMaxScaler()	Univariate Chi^2
	Bag of n-grams	MinMaxScaler()	Univariate Chi^2
	Bag-of-words	MinMaxScaler()	PCA
	Bag-of-words + Bag of n-grams	MinMaxScaler()	Univariate Chi^2
	TF-IDF	MaxAbsScaler()	Univariate Chi^2
Word Embeddings	Word2Vec from Word2Vec model	MinMaxScaler()	Univariate Chi^2
	Word2Vec from FastText model	MinMaxScaler()	Univariate Chi^2
	GloVe Embeddings with Flair	MinMaxScaler()	Univariate Chi^2

Scaling Used for the Features

- **MinMaxScaler**. Transform features by scaling each feature to a given range. This estimator scales and translates each feature individually such that it **is** in the given range on the training set, e.g. between zero and one. ... This transformation **is** often used as an alternative to zero mean, unit variance scaling.
- **MaxAbsScaler**. Scale each feature by its maximum absolute value. This estimator scales and translates each feature individually such that the maximal absolute value of each feature in the training set **will** be 1.0. It **does** not shift/center the data, and thus **does** not destroy any sparsity.

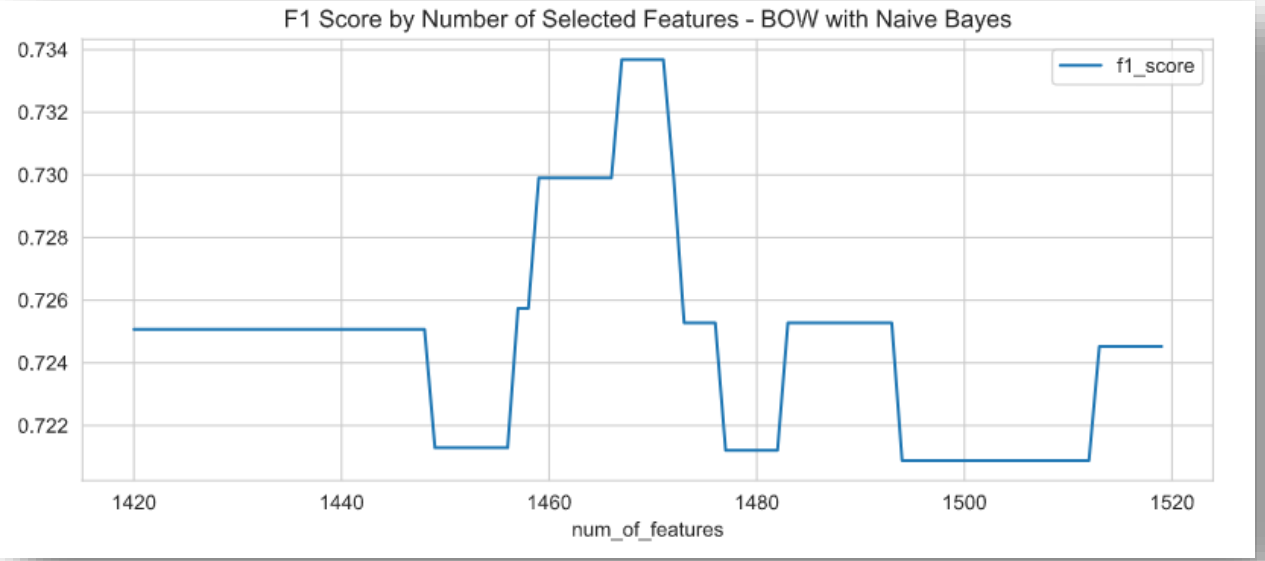
Bag of Words

Feature Extraction

	PAD	like	would	ok	okay	yes	want	pm	order	thank	time	tickets
0	0	3	3	3	0	1	1	1	0	0	0	0
1	0	0	0	3	0	0	2	0	0	0	1	0
2	0	1	0	0	2	3	1	5	0	2	0	5
3	0	0	0	0	3	0	1	5	0	1	3	0
4	0	2	1	0	2	2	2	0	3	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...
995	0	5	6	0	4	2	2	0	6	2	1	0
996	0	3	3	0	1	1	1	0	0	0	0	0
997	0	1	0	5	0	3	0	0	0	4	0	0
998	0	0	0	0	5	1	1	4	0	0	1	1
999	0	6	5	0	4	2	0	8	1	0	2	0

1000 rows × 6115 columns

Feature Selection – Univariate with Chi^2



Benchmarking

	Features_Benchedmarked	Feat_Type	Precision	Recall	f1_score	accuracy
0	BOW Naive Bayes Baseline	BOW	0.7243102	0.6960000	0.6838492	0.6960000
1	BOW Naive Bayes Optimal Features Selected: 1470	BOW	0.7656818	0.7360000	0.7336865	0.7360000

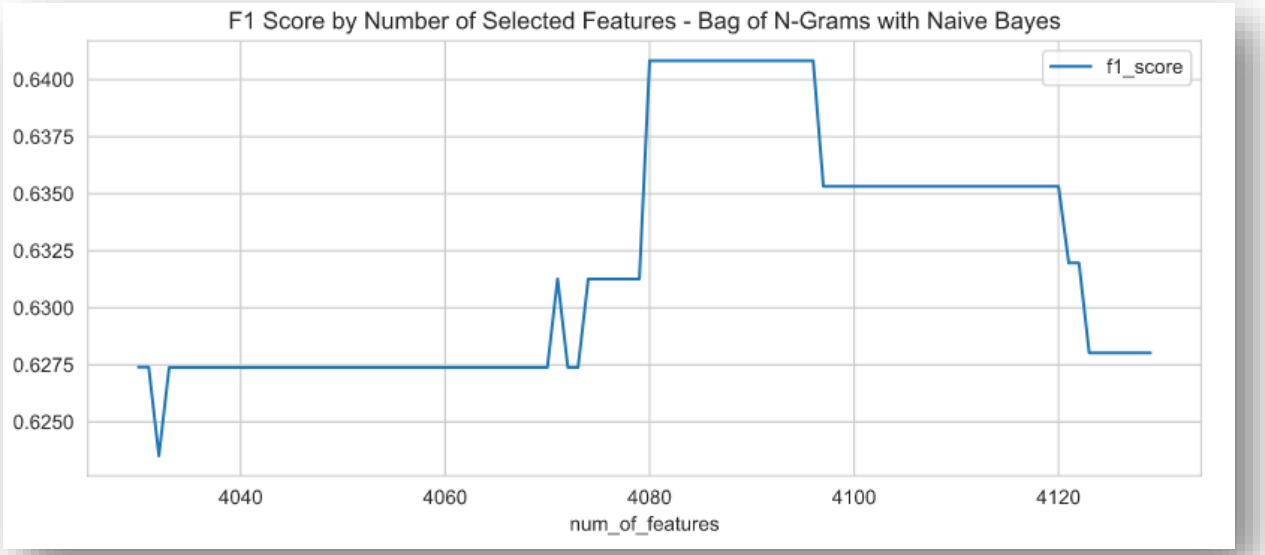
Bag of n-Grams

Feature Extraction

	abbey drive	abigail lives	abigails whoops	abigails yes	ability scan	able accommodate	able attend	able come	able find	able get	able make	able meet
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...
745	0	0	0	0	0	0	0	0	0	0	0	0
746	0	0	0	0	0	0	0	0	0	0	0	0
747	0	0	0	0	0	0	0	0	0	0	0	0
748	0	0	0	0	0	0	0	0	0	0	0	0
749	0	0	0	0	0	0	0	0	0	0	0	0

750 rows × 37534 columns

Feature Selection – Univariate with Chi^2



Benchmarking

	Features_Benchedmarked	Feat_Type	Precision	Recall	f1_score	accuracy
0	Bag of N-Gram Naive Bayes baseline	BONG	0.6401657	0.6000000	0.5697283	0.6000000
1	Bag of N-Gram Naive Bayes Optimal Features Selected: 4080	BONG	0.6797066	0.6480000	0.6408318	0.6480000

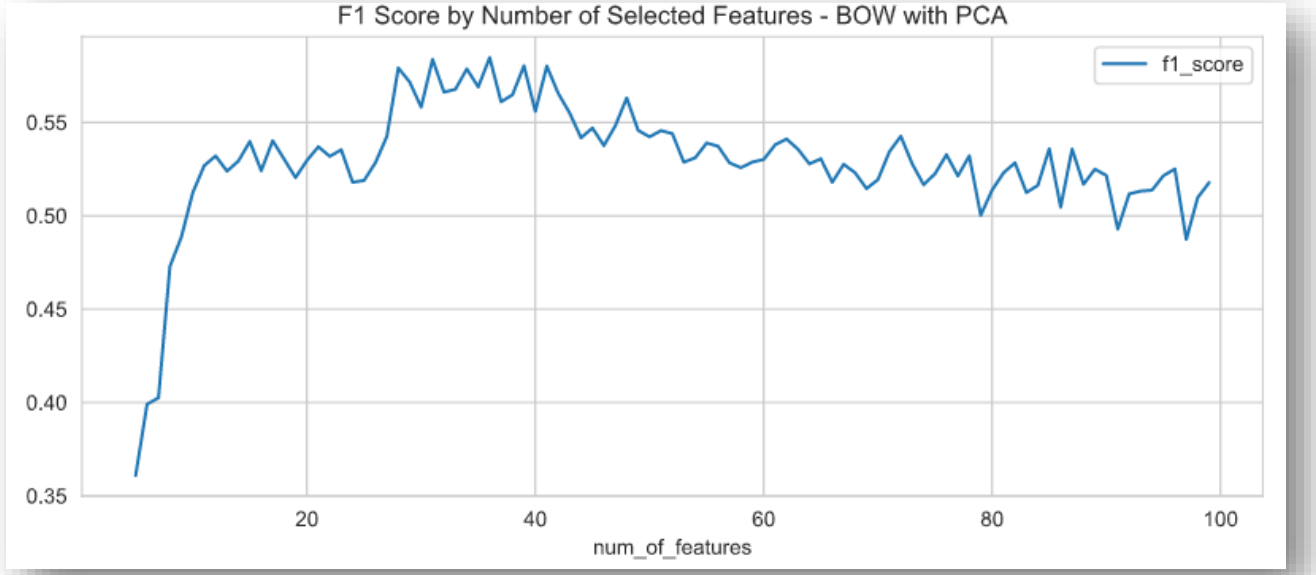
PCA on Bag of Words

Feature Extraction

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1.37	-1.61	-0.65	0.40	-2.47	-1.83	0.66	0.93	0.35	0.45	1.39	1.58	-0.88
1	-2.02	-2.62	0.33	-1.64	-0.99	-0.85	-2.02	0.87	2.24	-2.80	-0.42	0.22	-0.51
2	-3.09	3.23	1.07	-2.81	0.11	-0.45	-0.94	-0.00	-0.34	0.01	-1.27	0.09	-2.37
3	2.66	0.69	-2.27	2.91	-3.35	3.07	-1.85	-1.83	2.40	2.49	1.14	-0.56	-0.33
4	-2.38	-4.20	-1.04	0.02	-0.68	0.32	-0.51	-0.28	-0.03	-1.49	0.60	-0.13	0.46
...	...	...	...	...	...	...	...	...	...	...	...	...	...
745	-3.07	-0.57	0.11	-3.86	-0.67	-1.92	-1.83	0.87	1.02	0.05	-0.16	-0.69	0.69
746	-1.78	-1.48	2.56	1.75	-0.34	3.39	0.92	3.44	0.29	-1.81	-0.19	-0.40	0.18
747	-3.72	1.31	-4.59	-3.17	2.58	0.59	4.80	-1.17	0.02	2.34	0.15	0.58	-0.36
748	-1.90	-4.91	-2.57	-0.83	0.13	1.19	2.24	-2.61	0.83	0.23	0.04	-2.68	1.55
749	0.58	5.87	-5.46	-4.87	0.16	0.72	3.34	-0.25	1.99	3.47	0.84	-0.67	1.92

750 rows × 38 columns

Feature Selection – PCA



Benchmarking

	Features_Benchedmarked	Feat_Type	Precision	Recall	f1_score	accuracy
1	BOW With Top: 36 PCA Components Seleted	BOW_PCA	0.6207596	0.5920000	0.5804592	0.5920000

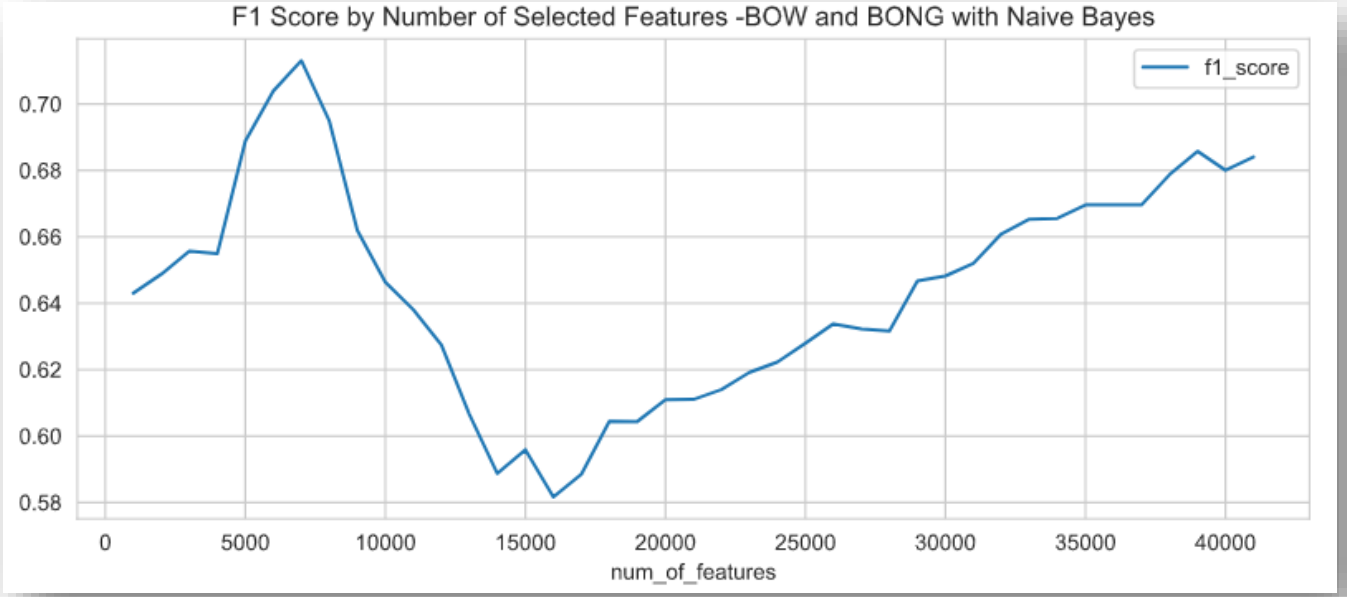
Combination – BOW and Bag of nGrams

Feature Extraction

	0	1	2	3	4	5
0	0.0000000	5.0000000	4.0000000	0.0000000	0.0000000	5.0000000
1	0.0000000	1.0000000	3.0000000	0.0000000	0.0000000	3.0000000
2	0.0000000	1.0000000	1.0000000	3.0000000	0.0000000	2.0000000
3	0.0000000	4.0000000	4.0000000	0.0000000	9.0000000	1.0000000
4	0.0000000	0.0000000	1.0000000	6.0000000	0.0000000	0.0000000
...	...	...	...	...	...	...
245	0.0000000	0.0000000	0.0000000	0.0000000	7.0000000	4.0000000
246	0.0000000	5.0000000	2.0000000	0.0000000	0.0000000	4.0000000
247	0.0000000	1.0000000	0.0000000	0.0000000	2.0000000	3.0000000
248	0.0000000	1.0000000	1.0000000	1.0000000	0.0000000	0.0000000
249	0.0000000	0.0000000	0.0000000	0.0000000	2.0000000	3.0000000

250 rows × 43649 columns

Feature Selection – Univariate with Chi^2



Benchmarking

	Features_Benchedmarked	Feat_Type	Precision	Recall	f1_score	accuracy
0	BOW + Bag of NGrams Top: 7000 Features with Naive Bayes	BOW_BONG	0.7848327	0.7200000	0.7130762	0.7200000

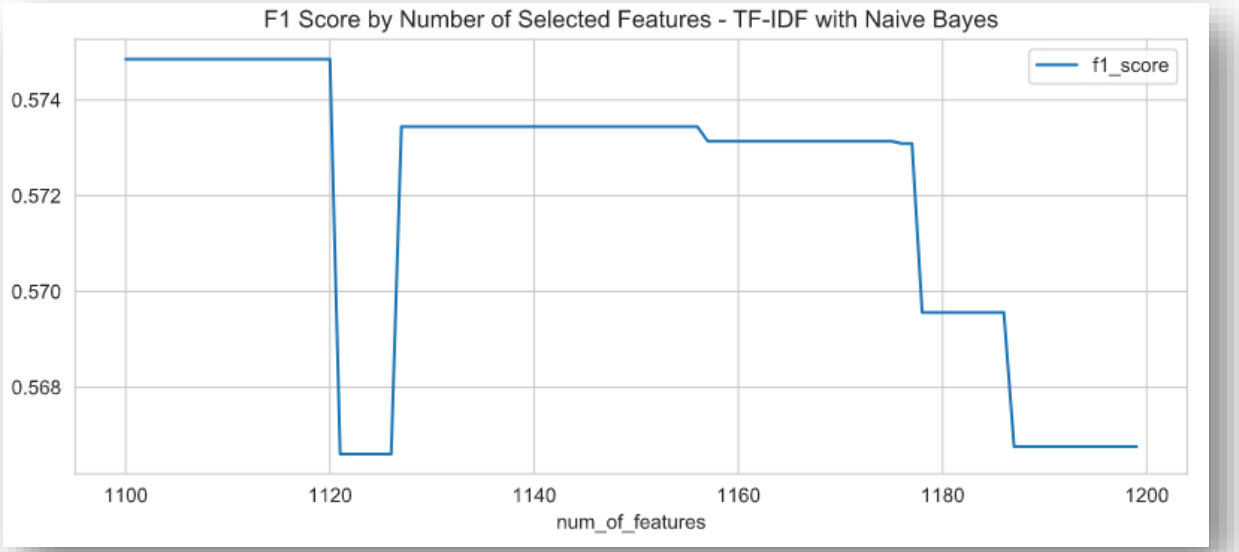
TF-IDF

Feature Extraction

	abbey	abigail	abigails	ability	able	abosolutly	abotu
0	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
1	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
2	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
3	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
4	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
...	...	...	...	...	...	...	...
745	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
746	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
747	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
748	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
749	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000

750 rows × 5221 columns

Feature Selection – Univariate with Chi^2



Benchmarking

	Features_Benchedmarked	Feat_Type	Precision	Recall	f1_score	accuracy
0	TF-IDF Naive Bayes Baseline	TF-IDF	0.6391206	0.6080000	0.5563710	0.6080000
1	TF-IDF Naive Bayes Optimal Features Selected: 1100	TF-IDF	0.7315170	0.6280000	0.5748536	0.6280000

Count Vectors

Word Cloud visualization of count vectors

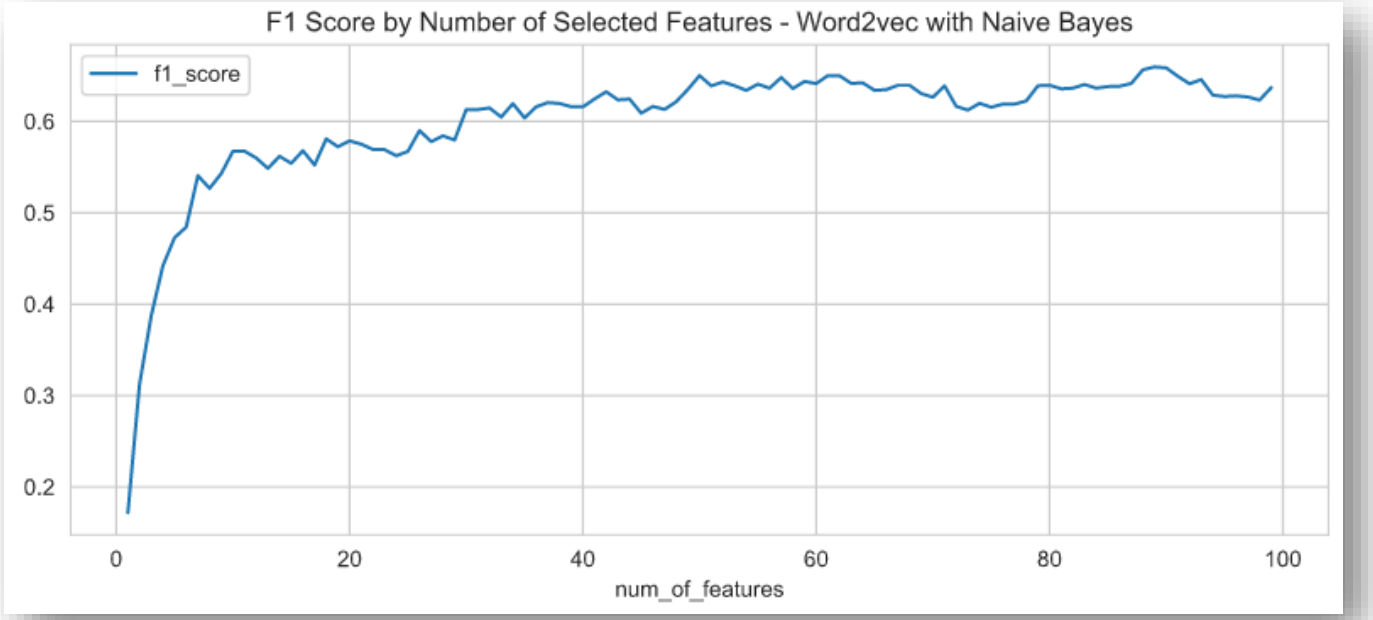
Word2Vec

Feature Extraction

	0	1	2	3	4	5
0	0.1485421	-0.7538016	-0.1134053	0.6662195	-0.7246261	-0.7991774
1	-0.0607553	1.0541956	0.4737130	-0.1523951	0.8053292	-0.4724531
2	1.0977465	0.2787353	0.1757089	0.1846928	0.1721655	-0.5292627
3	0.5198169	0.1662873	-0.9000216	-0.3820015	-0.5178834	0.2152452
4	0.9221550	0.6896073	-0.1255608	-0.7041550	-1.0570785	-0.2343613
...	...	...	...	...	...	...
745	0.7777841	0.2974014	0.0622340	0.0823392	0.1991102	-0.0638373
746	1.3327909	1.0493979	0.4724289	-0.1673032	-0.3365210	-0.0422698
747	1.1111339	0.1478708	0.4795337	0.4620506	0.4769705	-0.6561098
748	0.7482316	-0.0639104	1.0775760	0.2059366	-0.5521994	-0.7394230
749	1.1768541	-0.0455626	-0.4162890	0.0041162	0.6534415	-0.8599626

750 rows × 100 columns

Feature Selection – Univariate with Chi^2



Benchmarking

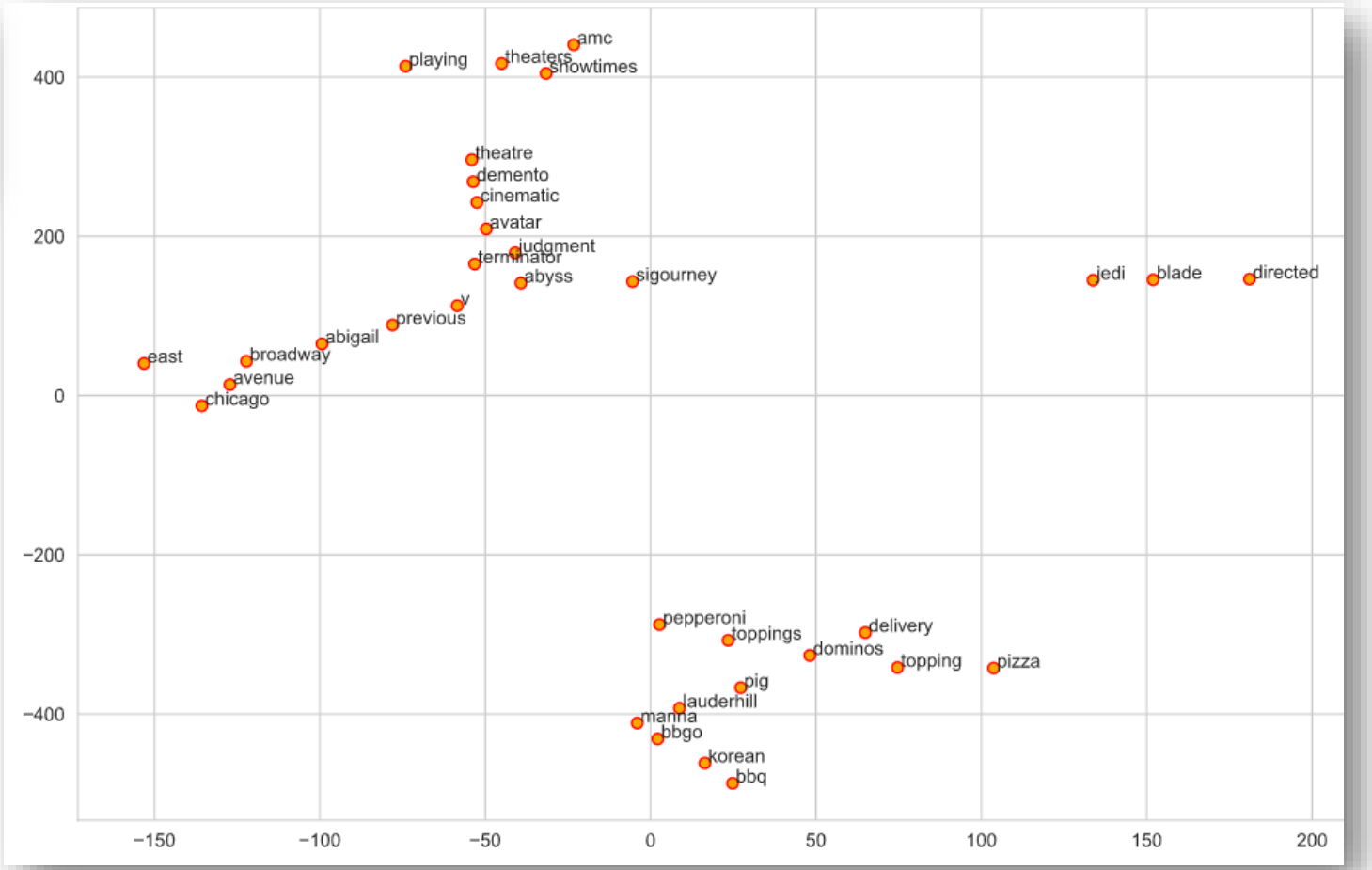
	Features_Benchedmarked	Feat_Type	Precision	Recall	f1_score	accuracy
0	Word2Vec Naive Bayes Baseline	Word2Vec	0.6512414	0.6480000	0.6379862	0.6480000
1	Word2Vec Naive Bayes Optimal Features Selected: 89	Word2Vec	0.6792014	0.6640000	0.6595766	0.6640000

Word2Vec – Word Embeddings

Word Embeddings

```
{'pizza': ['topping', 'dominos', 'toppings', 'delivery', 'pepperoni'],
 'terminator': ['judgment', 'abyss', 'avatar', 'directed', 'sigourney'],
 'star': ['wars', 'jedi', 'previous', 'v', 'blade'],
 'east': ['broadway', 'chicago', 'abigail', 'avenue', 'amc'],
 'korean': ['bbgo', 'lauderhill', 'bbq', 'manna', 'pig'],
 'playing': ['theaters', 'showtimes', 'theatre', 'cinematic', 'demento']}
```

Visualization of Word Embeddings



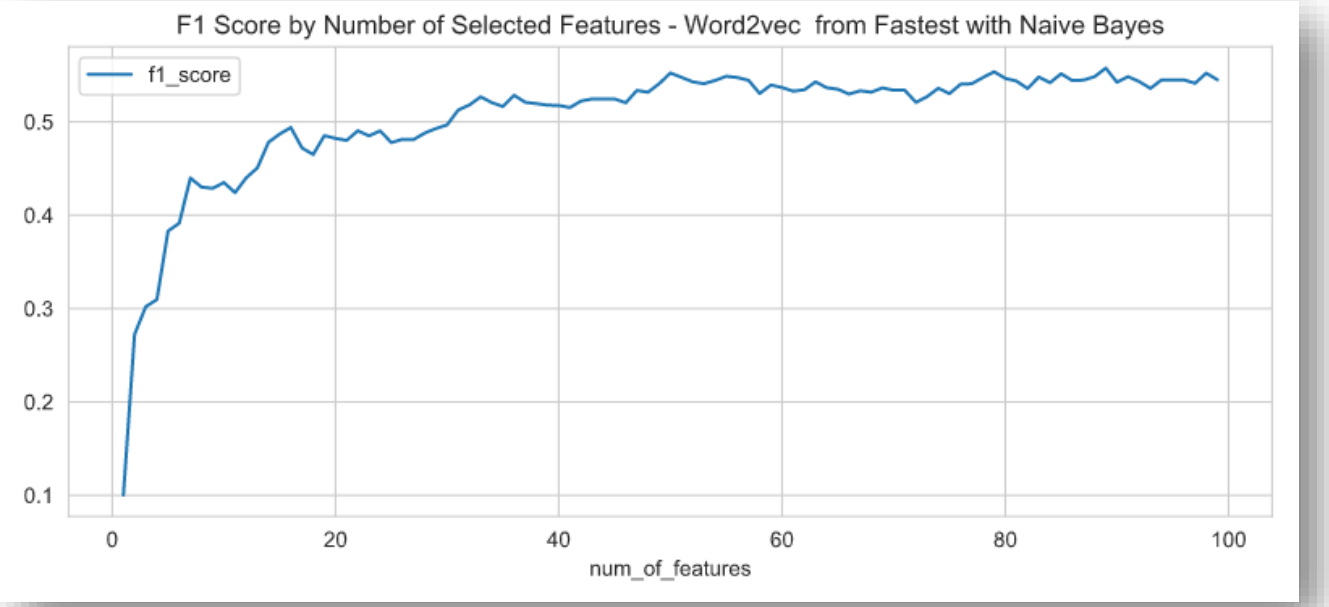
Word2Vec from FastText

Feature Extraction

	0	1	2	3	4	5
0	0.1485421	-0.7538016	-0.1134053	0.6662195	-0.7246261	-0.7991774
1	-0.0607553	1.0541956	0.4737130	-0.1523951	0.8053292	-0.4724531
2	1.0977465	0.2787353	0.1757089	0.1846928	0.1721655	-0.5292627
3	0.5198169	0.1662873	-0.9000216	-0.3820015	-0.5178834	0.2152452
4	0.9221550	0.6896073	-0.1255608	-0.7041550	-1.0570785	-0.2343613
...	...	...	...	...	...	...
745	0.7777841	0.2974014	0.0622340	0.0823392	0.1991102	-0.0638373
746	1.3327909	1.0493979	0.4724289	-0.1673032	-0.3365210	-0.0422698
747	1.1111339	0.1478708	0.4795337	0.4620506	0.4769705	-0.6561098
748	0.7482316	-0.0639104	1.0775760	0.2059366	-0.5521994	-0.7394230
749	1.1768541	-0.0455626	-0.4162890	0.0041162	0.6534415	-0.8599626

750 rows × 100 columns

Feature Selection – Univariate with Chi^2



Benchmarking

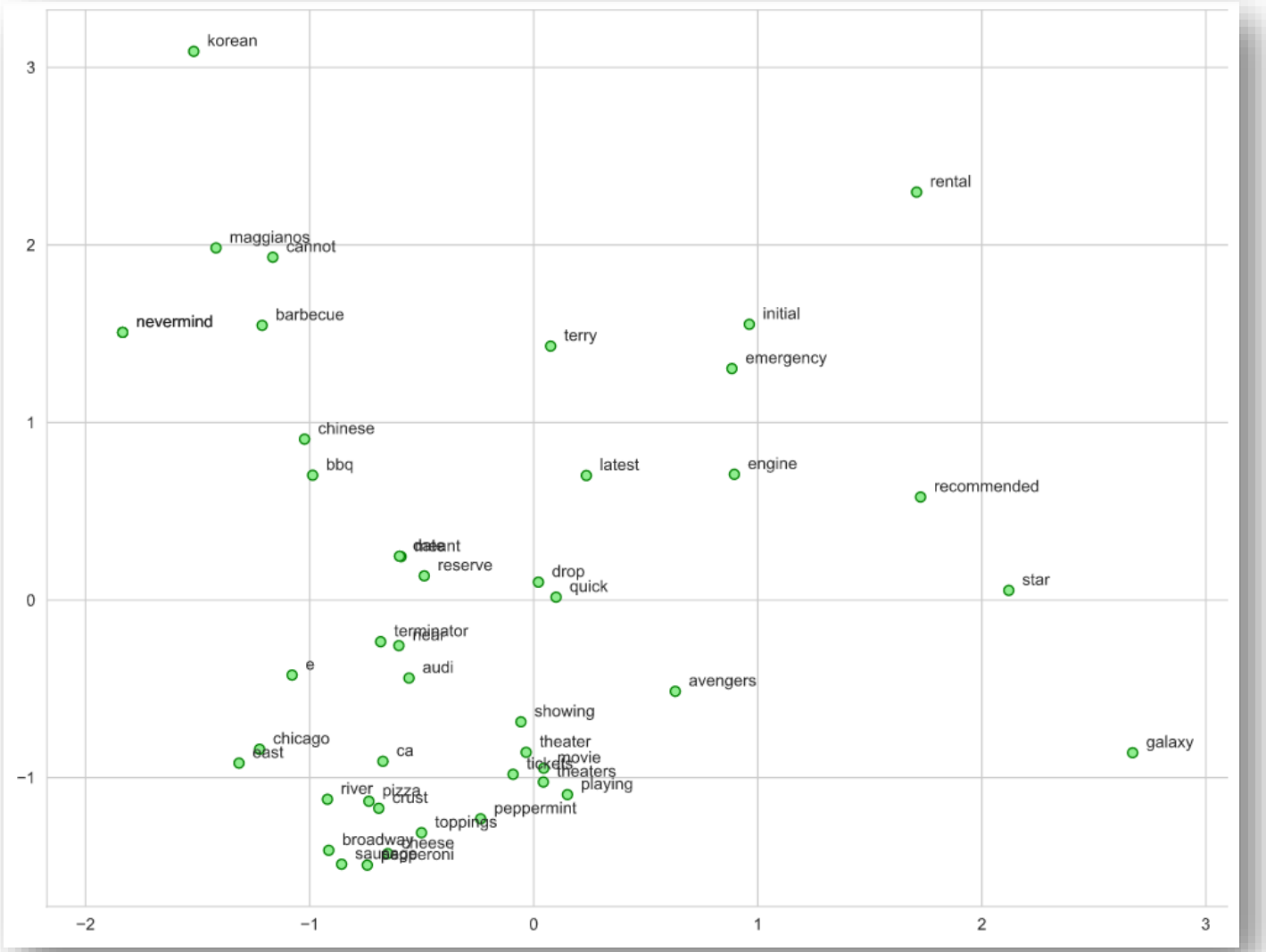
	Features_Benchedmarked	Feat_Type	Precision	Recall	f1_score	accuracy
0	Word2Vec Fastext Naive Bayes Baseline	Word2Vec_FT	0.6491872	0.5960000	0.5457620	0.5960000
1	Word2Vec from Fastest Naive Bayes Optimal Features Selected: 89	Word2Vec_FT	0.6491047	0.6080000	0.5575331	0.6080000

Word2Vec – FastText

Word Embeddings

```
{'rental': ['engine', 'emergency', 'quick', 'initial', 'drop'],
 'pizza': ['crust', 'pepperoni', 'toppings', 'cheese', 'sausage'],
 'terminator': ['nevermind', 'meant', 'near', 'peppermint', 'terry'],
 'star': ['wars', 'starring', 'galaxy', 'episode', 'recommended'],
 'audi': ['date', 'maggianos', 'reserve', 'avengers', 'latest'],
 'east': ['broadway', 'chicago', 'river', 'e', 'ca'],
 'korean': ['barbecue', 'bbq', 'cannot', 'chinese', 'nevermind'],
 'playing': ['tickets', 'movie', 'theater', 'showing', 'theaters']}
```

Visualization of Word Embeddings



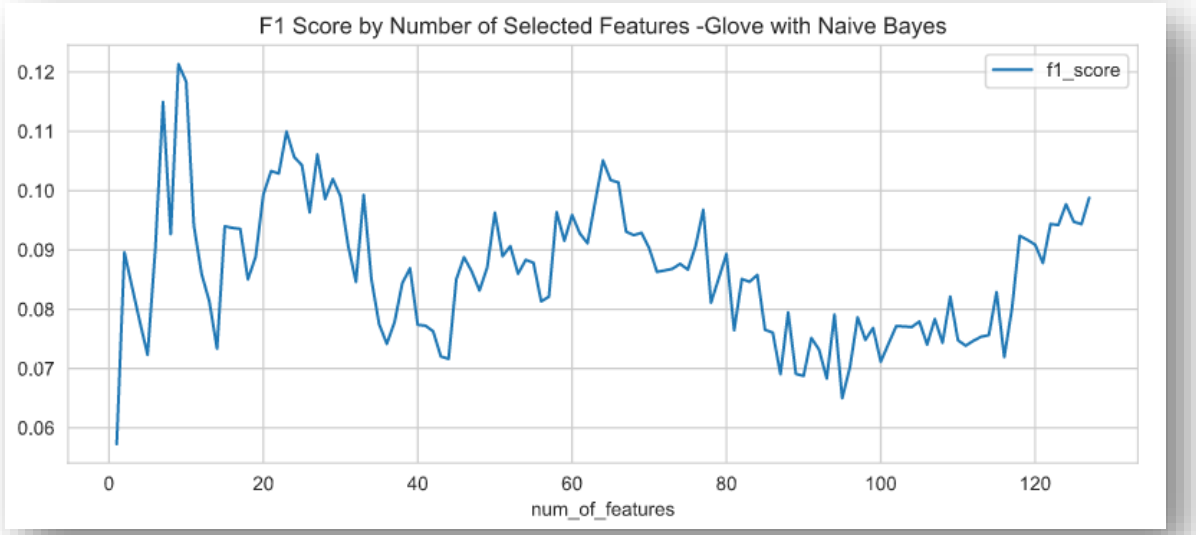
Glove from FLAIR

Feature Extraction

	0	1	2	3	4	5	6
0	-0.0117013	0.0617747	-0.0950755	0.1091563	-0.1017631	0.1596720	-0.0589565
1	0.0841807	-0.0620297	-0.1189875	-0.2875261	-0.0688366	0.0544598	-0.0740813
2	0.5025941	-0.0467734	-0.0588523	0.0163325	-0.1757609	-0.0487810	-0.0511439
3	-0.0227185	-0.0103282	-0.2404889	-0.1926294	-0.0662659	0.0504382	-0.2372182
4	-0.3222020	0.1958944	-0.4945650	-0.0861031	-0.0720596	0.1695689	-0.1488497
...	...	...	...	...	...	...	...
245	0.1313466	-0.2277337	-0.0699241	-0.1615932	-0.0721090	-0.0488684	-0.0867414
246	-0.2333377	-0.0675329	-0.3608953	-0.0170508	-0.2104710	0.1508308	-0.1324217
247	-0.1406810	0.1325267	-0.1478698	-0.1698386	-0.2022910	0.0391697	0.0031608
248	-0.1477484	-0.1279918	-0.3581616	-0.1365404	-0.1992545	-0.0465445	-0.2704552
249	0.4757498	-0.0056710	-0.0204031	0.0933946	-0.2478866	-0.0336119	0.0204238

250 rows × 128 columns

Feature Selection – Univariate with Chi^2



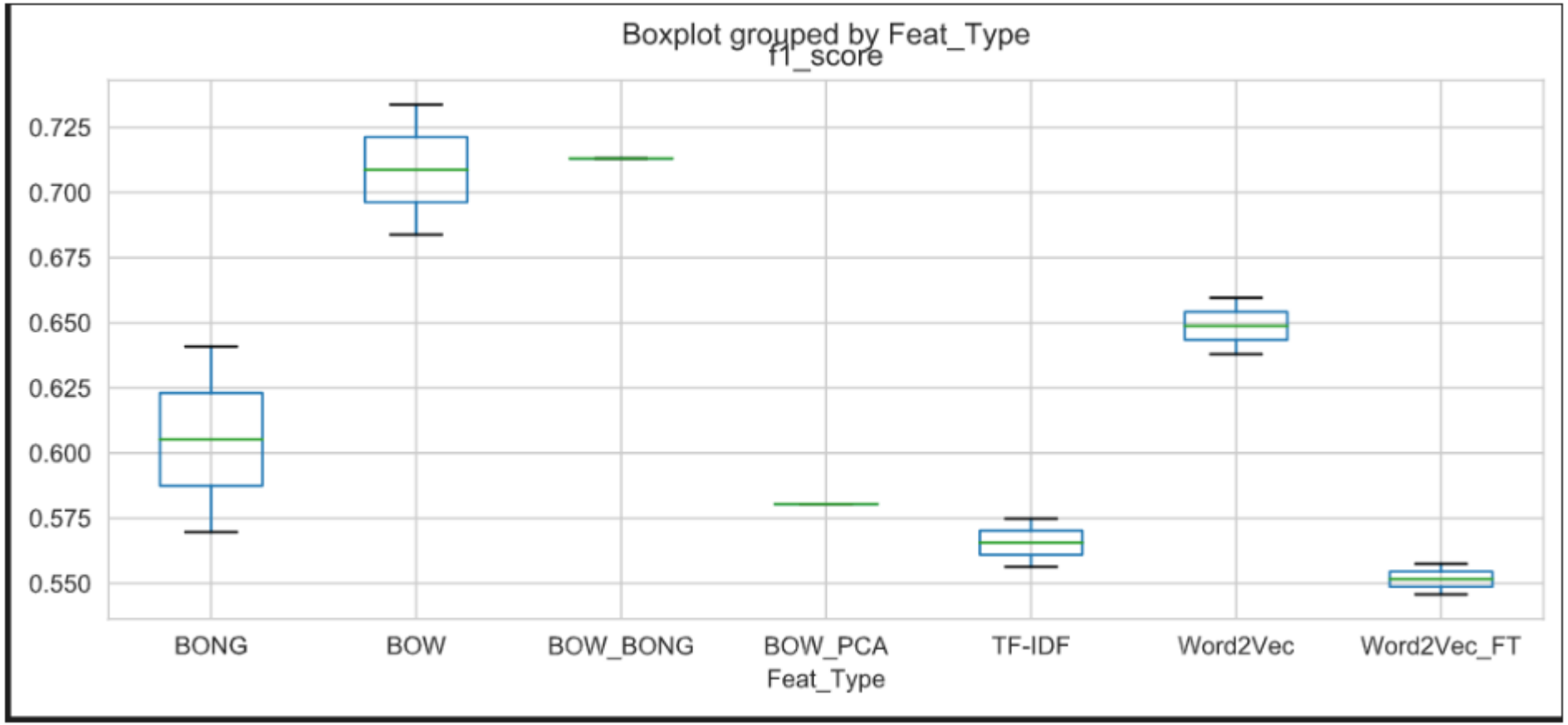
Benchmarking

	Features_Benchedmarked	f1_score	accuracy
0	Glove with Naive Bayes All Features	0.13	0.14

Feature Engineering, Extraction and Selection Final Results

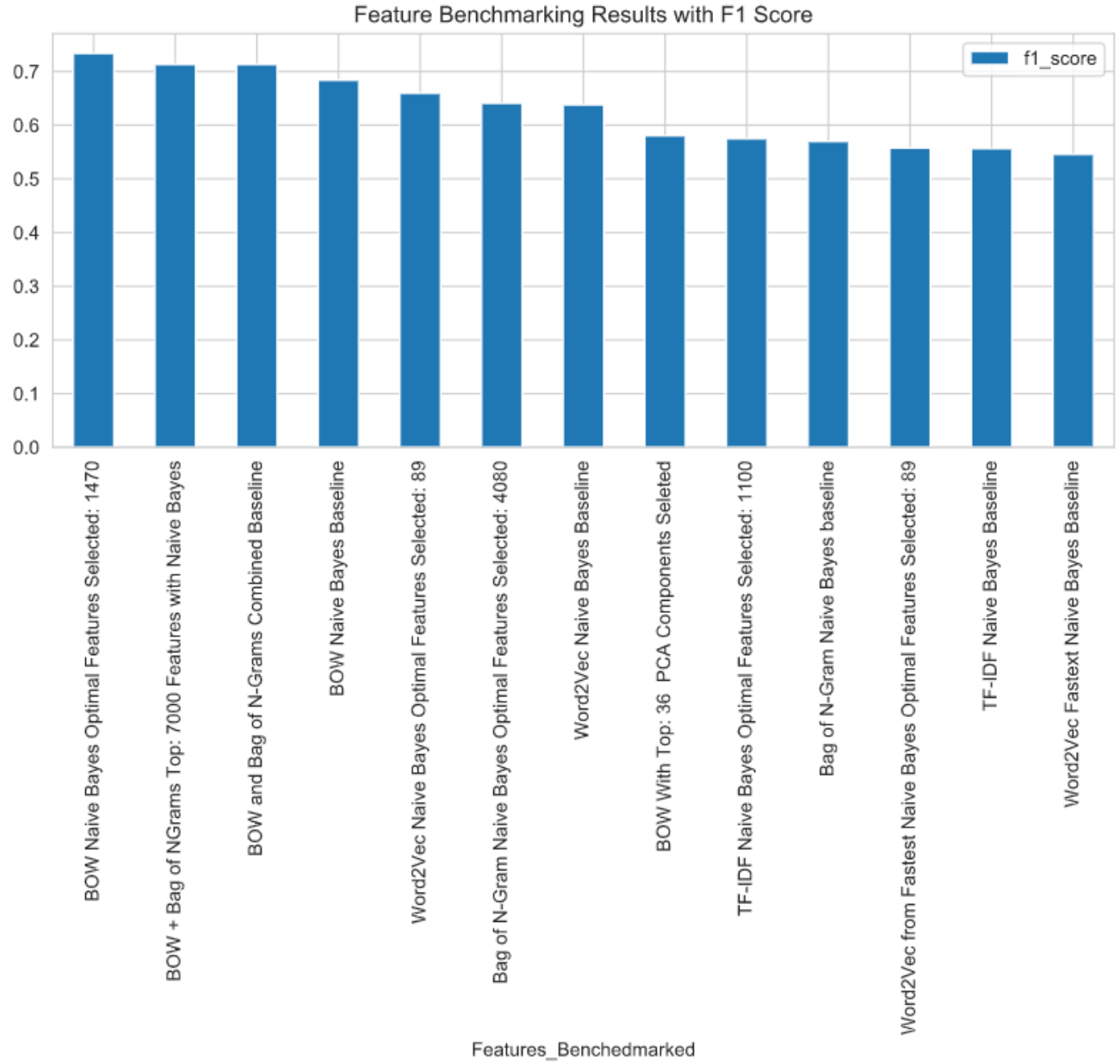
	Features_Benchedmarked	Feat_Type	Precision	Recall	f1_score	accuracy
0	BOW Naive Bayes Baseline	BOW	0.7243102	0.6960000	0.6838492	0.6960000
1	BOW Naive Bayes Optimal Features Selected: 1470	BOW	0.7656818	0.7360000	0.7336865	0.7360000
2	Bag of N-Gram Naive Bayes baseline	BONG	0.6401657	0.6000000	0.5697283	0.6000000
3	Bag of N-Gram Naive Bayes Optimal Features Selected: 4080	BONG	0.6797066	0.6480000	0.6408318	0.6480000
4	TF-IDF Naive Bayes Baseline	TF-IDF	0.6391206	0.6080000	0.5563710	0.6080000
5	TF-IDF Naive Bayes Optimal Features Selected: 1100	TF-IDF	0.7315170	0.6280000	0.5748536	0.6280000
6	Word2Vec Naive Bayes Baseline	Word2Vec	0.6512414	0.6480000	0.6379862	0.6480000
7	Word2Vec Naive Bayes Optimal Features Selected: 89	Word2Vec	0.6792014	0.6640000	0.6595766	0.6640000
8	Word2Vec Fastext Naive Bayes Baseline	Word2Vec_FT	0.6491872	0.5960000	0.5457620	0.5960000
9	Word2Vec from Fastest Naive Bayes Optimal Features Selected: 89	Word2Vec_FT	0.6491047	0.6080000	0.5575331	0.6080000
10	BOW + Bag of NGrams Top: 7000 Features with Naive Bayes	BOW_BONG	0.7848327	0.7200000	0.7130762	0.7200000
11	BOW and Bag of N-Grams Combined Baseline	BOW_BONG	0.7848327	0.7200000	0.7130762	0.7200000
12	BOW With Top: 36 PCA Components Seleted	BOW_PCA	0.6207596	0.5920000	0.5804592	0.5920000

Final Results –Variance Between Baseline and Optimized Features



Feature Engineering, Extraction and Selection Final Results

Visualization of Benchmark Results

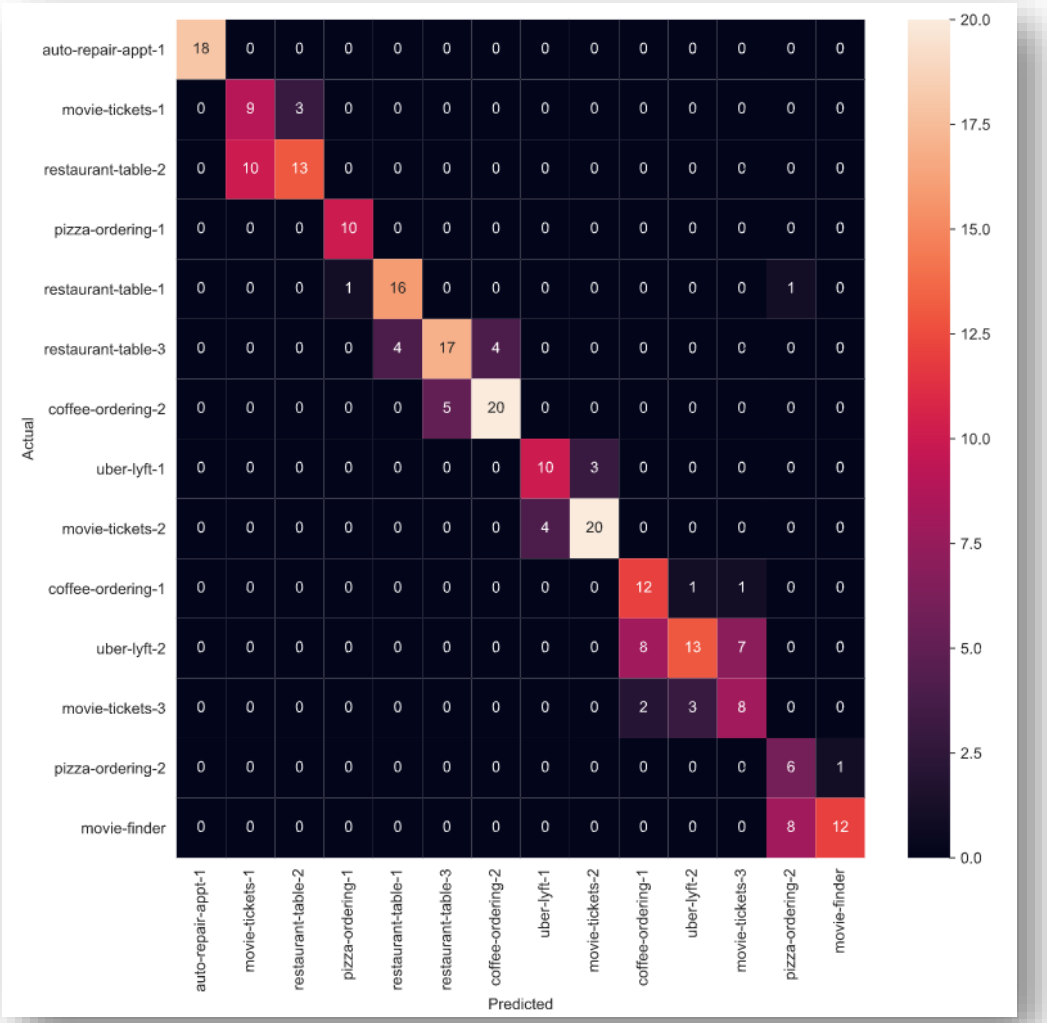


Feature Engineering, Extraction and Selection Final Results

Best results were produced from the following:	
Extracted Feature	Bag of Words
Feature Selection Method	Univariate chi^2
Reference Model	Naive Bayes Multinomial Variant

Label	precision	recall	f1-score	support
auto-repair-appt-1	1.00	1.00	1.00	18
coffee-ordering-1	0.75	0.47	0.58	19
coffee-ordering-2	0.57	0.81	0.67	16
movie-finder	1.00	0.91	0.95	11
movie-tickets-1	0.89	0.80	0.84	20
movie-tickets-2	0.68	0.77	0.72	22
movie-tickets-3	0.80	0.83	0.82	24
pizza-ordering-1	0.77	0.71	0.74	14
pizza-ordering-2	0.83	0.87	0.85	23
restaurant-table-1	0.86	0.55	0.67	22
restaurant-table-2	0.46	0.76	0.58	17
restaurant-table-3	0.62	0.50	0.55	16
uber-lyft-1	0.86	0.40	0.55	15
uber-lyft-2	0.60	0.92	0.73	13
accuracy			0.74	250
macro avg	0.76	0.74	0.73	250
weighted avg	0.77	0.74	0.73	250

Confusion Matrix Heatmap



Next Steps

- Feature Engineering
  - Review issue with Glove Feature in our notebook
  - Examine
    - Language Model (BERT and ELMO)
    - Topic Model
- Feature Selection Methods
  - Examine
    - RFE
    - Feature Importance