Indian Institute of Information Technology Surat

Lab Report on
Natural Language Processing (CS 601) Practical

Submitted by

[RAHUL KUMAR SINGH] (UI21CS44)

Course Faculty

Mrs. Nidhi Desai

Department of Computer Science and Engineering
Indian Institute of Information Technology Surat
Gujarat-394190, India

Jan-2024

Lab No: 5

Aim:

Perform n-gram operations on the dataset.For eg unigram, bi-gram and tri-gram. Analyze the outcomes of different n-gram and perform comparative Analysis.

Description:

- Objective: Perform n-gram operations (uni-gram, bi-gram, tri-gram) on text datasets and compare their outcomes.
- N-gram Extraction: Convert text into n-gram formats using TF-IDF vectorization.
- Datasets: Utilize NLTK datasets like Movie Reviews, Reuters Corpus, Twitter Samples, Product Reviews, Names Dataset, and Web Text Corpus.
- Comparative Analysis: Analyze and compare the accuracy, precision, recall, and F1-score of each n-gram model.
- Classification Task: Apply Logistic Regression for text classification.
- Outcome: Determine the optimal n-gram configuration based on dataset and classification task.

Source Code:

```
import nltk
from nltk.corpus import reuters, twitter_samples, product_reviews_1, names, webtext
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
import pandas as pd

nltk.download('movie_reviews')
nltk.download('twitter_samples')
nltk.download('names')

def prepare_movie_reviews():
    documents = [(list(movie_reviews.words(fileid)), category)
                 for category in movie_reviews.categories()
                 for fileid in movie_reviews.fileids(category)]
    reviews = [' '.join(doc) for doc, _ in documents]
    labels = [1 if label == 'pos' else 0 for _, label in documents]
    return reviews, labels

def prepare_twitter_samples():
    positive_tweets = twitter_samples.strings('positive_tweets.json')
    negative_tweets = twitter_samples.strings('negative_tweets.json')
    reviews = positive_tweets + negative_tweets
    labels = [1] * len(positive_tweets) + [0] * len(negative_tweets)
    return reviews, labels

datasets = [
    ('Movie Reviews', prepare_movie_reviews),
    ('Twitter Samples', prepare_twitter_samples),
]
```

```
def evaluate_model(ngram_range, ngram_type, X_train, X_test, y_train, y_test):
    vectorizer = TfidfVectorizer(ngram_range=ngram_range)
    X_train_vec = vectorizer.fit_transform(X_train)
    X_test_vec = vectorizer.transform(X_test)
    clf = LogisticRegression(max_iter=1000)
    clf.fit(X_train_vec, y_train)
    y_pred = clf.predict(X_test_vec)
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='weighted')
    recall = recall_score(y_test, y_pred, average='weighted')
    f1 = f1_score(y_test, y_pred, average='weighted')

    return {
        'ngram_type': ngram_type,
        'accuracy': accuracy,
        'precision': precision,
        'recall': recall,
        'f1_score': f1
    }

for dataset_name, dataset_fn in datasets:
    print(f"Evaluating on {dataset_name} dataset...")
    reviews, labels = dataset_fn()
    dataset_df = pd.DataFrame({'text': reviews, 'label': labels})
    print(dataset_df.head())
    X_train, X_test, y_train, y_test = train_test_split(reviews, labels, test_size=0.3,
random_state=42)
    results = []
    results.append(evaluate_model((1, 1), 'Uni-grams', X_train, X_test, y_train, y_test))
    results.append(evaluate_model((1, 2), 'Uni + Bi-grams', X_train, X_test, y_train, y_test))
    results.append(evaluate_model((1, 3), 'Uni + Bi + Tri-grams', X_train, X_test, y_train,
y_test))
    print("Evaluation Metrics: ")
    results_df = pd.DataFrame(results)
    print(results_df)
```

Output:

n-gram:

	text	unigrams	bigrams	trigrams
0	IF YA SMELL..... @TheRock has come back to #W...	[(if,), (ya,), (smell,), (.....,), (@,), (ther...	[(if, ya), (ya, smell), (smell, ...), (.....	[(if, ya, smell), (ya, smell, ...), (smell, ...
1	Who had the best Instagram photo of the week?!...	[(who,), (had,), (the,), (best,), (instagram,)...	[(who, had), (had, the), (the, best), (best, i...	[(who, had, the), (had, the, best), (the, best...
2	These #RoyalRumble crashers were RUTHLESS! ht...	[(these,), (#,), (royalrumble,), (crashers,), ...	[(these, #), (#, royalrumble), (royalrumble, c...	[(these, #, royalrumble), (#, royalrumble, cra...
3	An All Mighty moment in the 2023 Men's #RoyalR...	[(an,), (all,), (mighty,), (moment,), (in,), (...	[(an, all), (all, mighty), (mighty, moment), (...	[(an, all, mighty), (all, mighty, moment), (mi...
4	Outta nowhere! 😳	[(outta,), (nowhere,), (!,), (😳,)]	[(outta, nowhere), (nowhere, !), (!, 😳)]	[(outta, nowhere, !), (nowhere, !, 😳)]

Top 10 Uni-grams:

	N-gram	Count
0	@	46
1	#	46
2	!	36
3	:	21
4	wwe	18
5	wwenxt	17
6	.	14
7	,	14
8	?	12
9	https	9

Top 10 Bi-grams:

	N-gram	Count
0	# wwenxt	17
1	https :	9
2	, @	7
3	. #	6
4	oba femi	6
5	north american	6
6	american champion	5
7	# royalrumble	4
8	judgment day	4
9	! #	4

Top 10 Tri-grams:

	N-gram	Count
0	north american champion	5
1	. # wwenxt	4
2	# royalrumble match	3
3	8/7c @ usanetwork	3
4	: https :	3
5	biography : wwe	3
6	: wwe legends	3
7	! https :	2
8	https : //tube.mint.lgbt/vv5fxhfxce4	2
9	: //tube.mint.lgbt/vv5fxhfxce4 ?	2

Comparative Analysis on WWE Twitter Dataset:

Evaluating on WWE twitter dataset...

	text	label
0	IF YA SMELL..... @TheRock has come back to #W...	1
1	Who had the best Instagram photo of the week?!...	0
2	These #RoyalRumble crashers were RUTHLESS! ht...	0
3	An All Mighty moment in the 2023 Men's #RoyalR...	0
4	Outta nowhere! 😳	1

Evaluation Metrics:

	ngram_type	accuracy	precision	recall	f1_score
0	Uni-grams	0.600000	0.444444	0.8	0.571429
1	Uni + Bi-grams	0.733333	0.555556	1.0	0.714286
2	Uni + Bi + Tri-grams	0.666667	0.500000	1.0	0.666667
3	Bi-grams	0.600000	0.454545	1.0	0.625000
4	Tri-grams	0.600000	0.454545	1.0	0.625000
5	Bi + Tri-grams	0.600000	0.454545	1.0	0.625000

Comparative Analysis on Movie Reviews Dataset:

```
Evaluating on Movie Reviews dataset...
                                                text  label
0  plot : two teen couples go to a church party ,...      0
1  the happy bastard ' s quick movie review damn ...      0
2  it is movies like these that make a jaded movi...      0
3  " quest for camelot " is warner bros . ' first...      0
4  synopsis : a mentally unstable man undergoing ...      0
Evaluation Metrics:
             ngram_type  accuracy  precision    recall  f1_score
0             Uni-grams  0.810000   0.810387  0.810000  0.809968
1       Uni + Bi-grams  0.770000   0.784943  0.770000  0.767188
2  Uni + Bi + Tri-grams  0.730000   0.776153  0.730000  0.718769
3              Bi-grams  0.821667   0.824398  0.821667  0.821359
4             Tri-grams  0.796667   0.798685  0.796667  0.796395
5       Bi + Tri-grams  0.808333   0.816280  0.808333  0.807252
```

Comparative Analysis on Twitter Samples Dataset:

```
Evaluating on Twitter Samples dataset...
                                                text  label
0  #FollowFriday @France_Inte @PKuchly57 @Milipol...      1
1  @Lamb2ja Hey James! How odd :/ Please call our...      1
2  @DespiteOfficial we had a listen last night :)...      1
3                              @97sides CONGRATS :)      1
4  yeaaaah yippppy!!!  my accnt verified rqst has...      1
Evaluation Metrics:
             ngram_type  accuracy  precision    recall  f1_score
0             Uni-grams  0.765667   0.766495  0.765667  0.765676
1       Uni + Bi-grams  0.770333   0.771337  0.770333  0.770327
2  Uni + Bi + Tri-grams  0.768667   0.769603  0.768667  0.768667
3              Bi-grams  0.692000   0.705070  0.692000  0.688578
4             Tri-grams  0.607333   0.675957  0.607333  0.571347
5       Bi + Tri-grams  0.687667   0.702358  0.687667  0.683613
```

Conclusion:

- WWE Twitter Dataset: (Uni + Bi) n-grams yield optimal performance.
- Movie Reviews Dataset: Bi-grams provide the best results.
- Twitter Samples Dataset: (Uni + Bi) n-grams achieve top accuracy.
- Performance varies across datasets making it essential to optimize n-grams.