Quantum-Enhanced Natural Language Processing for Sentiment Analysis

AI Research Team

September 13, 2024

Abstract

This paper presents a sophisticated approach to sentiment analysis by integrating classical NLP techniques with quantum-inspired algorithms. The objective is to enhance the performance and accuracy of sentiment analysis models using quantum computing principles.

1 Introduction

Sentiment analysis is a crucial task in natural language processing (NLP) that involves classifying text into positive or negative sentiment. Traditional methods rely on classical machine learning algorithms, but recent advancements in quantum computing offer new opportunities for enhancement.

2 Data Preprocessing

Data preprocessing involves cleaning the text data to ensure it is suitable for analysis. This includes converting text to lowercase, removing short words, URLs, and non-alphanumeric characters.

Listing 1: Data Preprocessing

```
import pandas as pd
import re

df = pd.read_csv('sentiment140.csv', encoding='latin1', header=None)
df.columns = ['target', 'id', 'date', 'flag', 'user', 'text']

def preprocess_text(text):
    text = text.lower()
    text = re.sub(r'\b\w{1,2}\b', '', text)
    text = re.sub(r'http\S+', '', text)
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    return text
```

```
df['text'] = df['text'].apply(preprocess_text)
df['target'] = df['target'].apply(lambda x: 0 if x == 0 else 1)
```

3 Classical NLP Techniques

We use TF-IDF vectorization to convert text data into numerical features and a Naive Bayes classifier for sentiment analysis.

Listing 2: Classical NLP Model

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report

X_train, X_test, y_train, y_test = train_test_split(df['text'], df['target'], te

vectorizer = TfidfVectorizer(max_features=3000)
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

nb_classifier = MultinomialNB()
nb_classifier.fit(X_train_tfidf, y_train)
y_pred_classical = nb_classifier.predict(X_test_tfidf)

print("Classical␣Model␣Accuracy:", accuracy_score(y_test, y_pred_classical))
print(classification_report(y_test, y_pred_classical))
```

4 Quantum-Inspired Enhancements

A quantum circuit is created using Qiskit to simulate quantum feature extraction, which is combined with classical features to enhance the model.

Listing 3: Quantum-Enhanced Features

```
from qiskit import Aer, QuantumCircuit, transpile, assemble, execute
from qiskit.visualization import plot_histogram
import numpy as np

def quantum_enhanced_analysis(text):
    quantum_circuit = QuantumCircuit(2, 2)
    quantum_circuit.h(0)
    quantum_circuit.cx(0, 1)
    quantum_circuit.measure([0, 1], [0, 1])
    simulator = Aer.get_backend(’qasm_simulator’)
    transpiled_circuit = transpile(quantum_circuit, simulator)
    qobj = assemble(transpiled_circuit)
    result = execute(transpiled_circuit, backend=simulator).result()
    counts = result.get_counts()

    feature = int(counts.get(’00’, 0)) / sum(counts.values())
    return feature

X_train_quantum = np.array([quantum_enhanced_analysis(text) for text in X_train]
X_test_quantum = np.array([quantum_enhanced_analysis(text) for text in X_test])

X_train_combined = np.hstack((X_train_tfidf.toarray(), X_train_quantum.reshape(−
X_test_combined = np.hstack((X_test_tfidf.toarray(), X_test_quantum.reshape(−1,

nb_classifier_enhanced = MultinomialNB()
nb_classifier_enhanced.fit(X_train_combined, y_train)
y_pred_enhanced = nb_classifier_enhanced.predict(X_test_combined)

print(”Quantum−Enhanced␣Model␣Accuracy:”, accuracy_score(y_test, y_pred_enhanced
print(classification_report(y_test, y_pred_enhanced))

plot_histogram(counts)
```

5 Conclusion

This paper demonstrates the potential of integrating quantum computing principles with classical NLP techniques to enhance the performance of sentiment analysis models. The quantum-inspired enhancements show promising results, indicating a new direction for future research in NLP.