Comparative Analysis of N-grams, Large Language Model, and Embedding Techniques in Sentiment Analysis

Noravee Kanchanavatee

Findings

- Accuracy of sentimental analysis on yelp's reviews with scores from 1, most negative, to 5, most positive:

Technique	Accuracy
N-grams (N = 1,2) → TF-IDF → XGBoost	0.569
Zero-shot LLM (Gemini)	0.586
Embedding (Gemini) → XGBoost	0.675

- Each model can classify 1 and 5 well but struggles with the rest.
- Zero-shot LLM classify samples to mostly class 1 and 5 while techniques that converting texts to vector before classification generate more balanced predictions.

Introduction

Sentiment analysis, also known as opinion mining, is a powerful tool that businesses use to understand customer perceptions and opinions. This analysis involves examining text data from sources like social media posts, reviews,

and customer feedback to uncover how customers feel about a product, service, or brand.

By combining sentiment analysis with other Natural Language Processing (NLP) tasks like topic identification, businesses can gain deeper insights. Topic identification helps discover the main themes in a text, allowing businesses not only to understand sentiments but also to associate them with specific topics.

Moreover, sentiment analysis is valuable for applications such as recommendation systems. Analyzing user sentiments in comments or reviews helps recommendation systems understand user preferences better, improving their performance.

The emergence of Large Language Models (LLMs) has significantly improved sentiment analysis quality. LLMs understand context, sarcasm, and complex language constructs, leading to more accurate sentiment analysis. Comparing traditional NLP techniques like bag of words to LLM-based approaches is essential to understand their strengths and limitations.

Overall, studying and comparing different techniques for sentiment analysis is crucial for businesses. It helps identify strengths and weaknesses, monitor brand reputation, understand customer needs, and make data-driven decisions. These insights are valuable for improving customer service, guiding product development, and ultimately driving business growth and success in today's digital landscape.

Methodology

The data used in this study is constructed by Xiang Zhang from the Yelp Dataset Challenge 2015 data. It is first used as a text classification benchmark in the following paper: Xiang Zhang, Junbo Zhao, Yann LeCun. Character-level Convolutional Networks for Text Classification. Advances in Neural Information Processing Systems 28 (NIPS 2015). There are 650,000 training samples and 50,000 testing samples. Each set has the same number review star from 1 to 5. Only 2 columns are in the data set, corresponding to class index (1 to 5) and review texts. The texts are escaped using double quotes ("), and any internal double quote is escaped by 2 double quotes (""). New lines are escaped by "\n".

In this study, two main methods are considered.

1. Convert texts into vectors and then apply XGBoost for classsification.

Vectorization were done by two different techniques:

- N-grams and term frequency–inverse document frequency (TF-IDF)
- Embedding using Google Gemini embeddding-001 model which turns texts into vector with 768 dimensions

These models were first fitted to the training data before used to predict the outcome from the testing set. Hyperparameter optimization was also performed on the XGBoost using baysian optimizaiton with 3-fold cross validation (CV) accuracy score as a metric. The embedding model was further optimized by principal component analysis (PCA).

2. Directly use raw texts as input for the LLM without any example (zero-shot). The LLM used in this study is Google gemini-1.0-pro with temperature or randomness set to zero for repeatibility and all safety filter off since some reviews contain a multitude of inappropiate words. Even without safety blocking, the model still fails to generate response for a few samples. In those extreme cases, the texts were sliced off from the end by 10% each time the model do not respond.

   The prompt or input texts for the task are as followed:

   "Classify sentiment from 1-5 (negative-positive) for the following statement. Respond with a single digit, 1 for the most negative, 2 for slightly negative, 3 for neutral, 4 for slightly positive, and 5 for the most positive."

   This method does not require any training data. The accuracy was measured on the test set only.

Results

N-grams and TF-IDF

Several models using different grams or number of consecutive words were performed. As can be seen in Figure 1, the best model according to the cv score is the one with unigram and bigram. Accuracy on the test set of the N = 1 and 2 model is 0.569. Figure 2 shows the confusion matrix of the test data. Frequency of the predicted class 1 and 5 are slightly higher. They

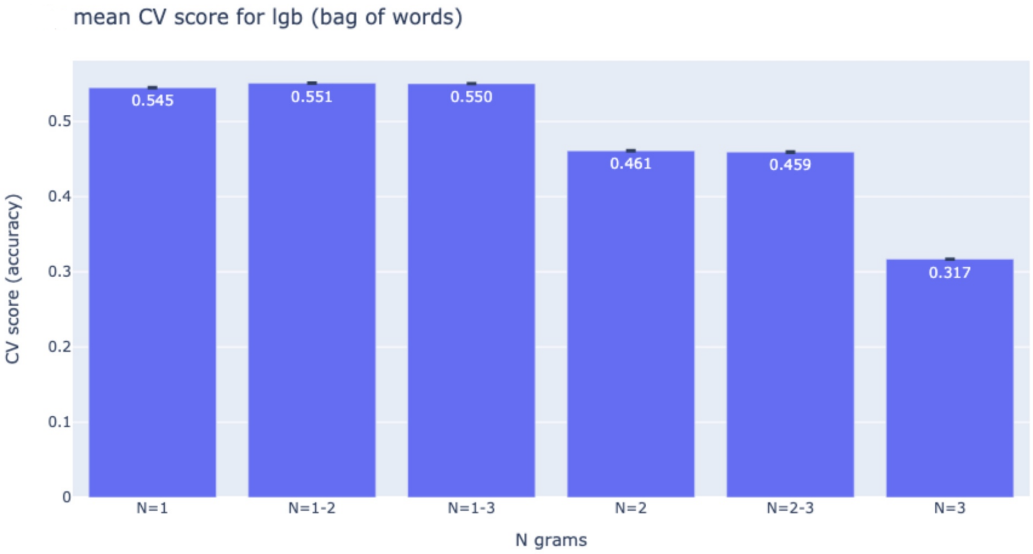


Figure 1: Number of sequence of words and CV score

also have better precision and recall than class 2,3, and 4. The differences between precision and recall for each class is not large, and F1-scores are approximately 0.5 and 0.7 for class 2,3,4 and 1,5, respectively.

Zero-shot LLM

The test accuracy for zero-shot LLM is 0.586, slightly higer than that of traditional NLP technique. It can be seen from Figure 3 that more than 60% of the predicted values are 1 and 5 so the recalls of those classes are very high. On the contary, less than 10% of the prediction is 3, but its precision is quite high at 70%. Thus, there is a big gap between precision and recall in each class, especially 1,3, and 5. This model has slightly better F1-score on class 1 and 5 and worse on class 2,3,4 when compared to the previous model.

Embeddings

This model yields test accurac of 0.671. Prediction from this model has quite balanced classes and overall better performance compared to the N-gram and zero-shot LLM (Figure 4(a)). Slight improvement on accuracy can

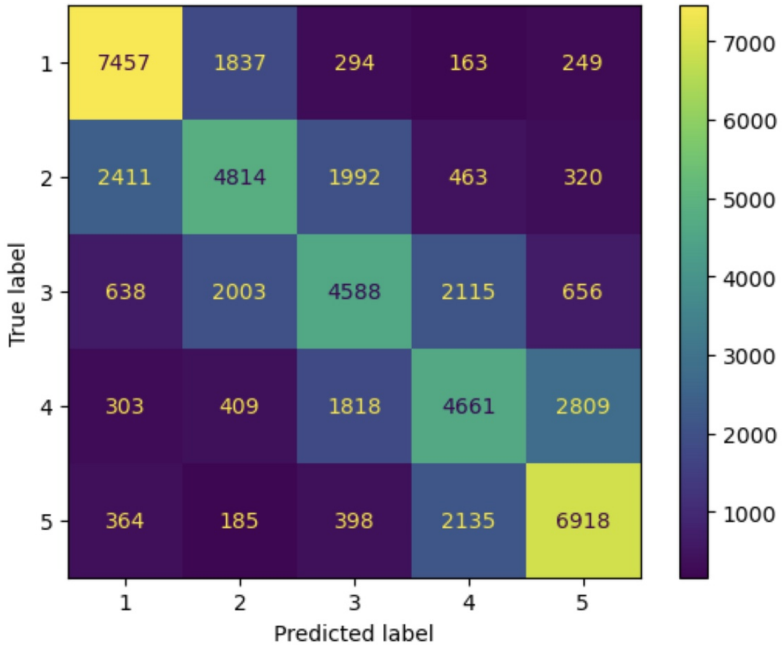


Figure 2: Confusion matrix of N-grams and TF-IDF model

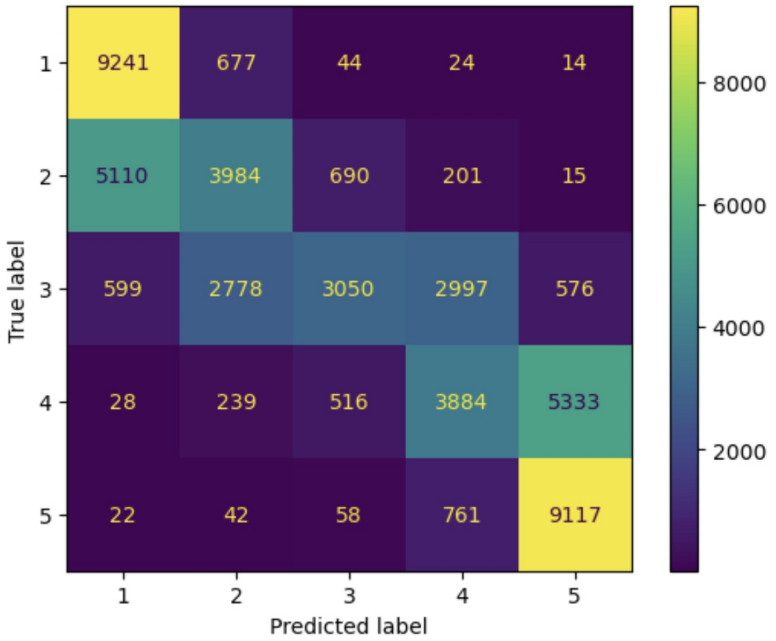
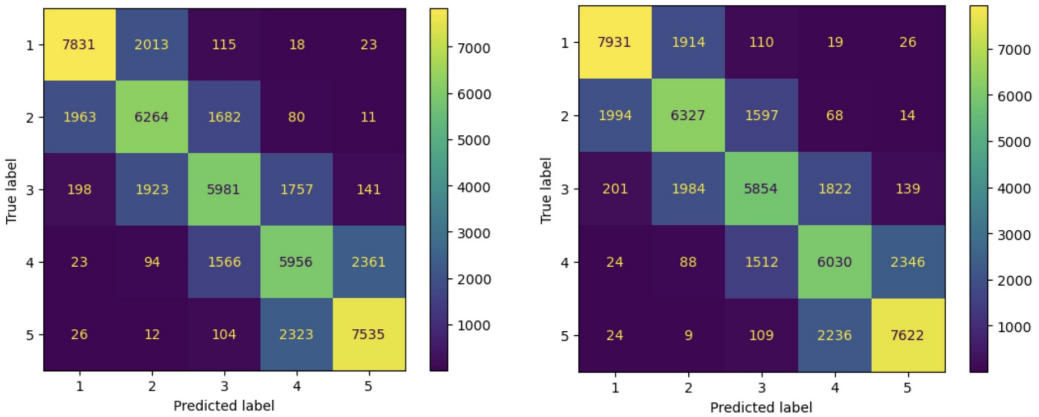


Figure 3: Confusion matrix of zero-shot LLM model



(a) (b)

Figure 4: Confusion matrix of (a) embedding model (b) embedding model after PCA

be achieved by PCA with $n = 528$ where the number of principal component, n, was choosen based on CV score. Note that $n = 528$ did not lead to the highest CV score but rather where the score starts to plateau. The result of this optimization can be seen in Figure 4(b), which is similar to previous model with slightly increase in accuracy at 0.675. This enhancement is due to better performance in every class except class 3.

Model comparison

Figure 5 illustrates test accuracy of N-gram, zero-shot LLM, and embedding models. While the embedding model shows the highest accuracy for each class, the zero-shot LLM has even higher accuracy for class 1 and 5 at the cost of poor performance on the rest. Both techniques of converting texts to vectors result in a similar pattern of the confusion matrix, approximately 70-80% for class 1 and 5, around 45-60% for the rest, with the embedding model having the higher score in both cases.

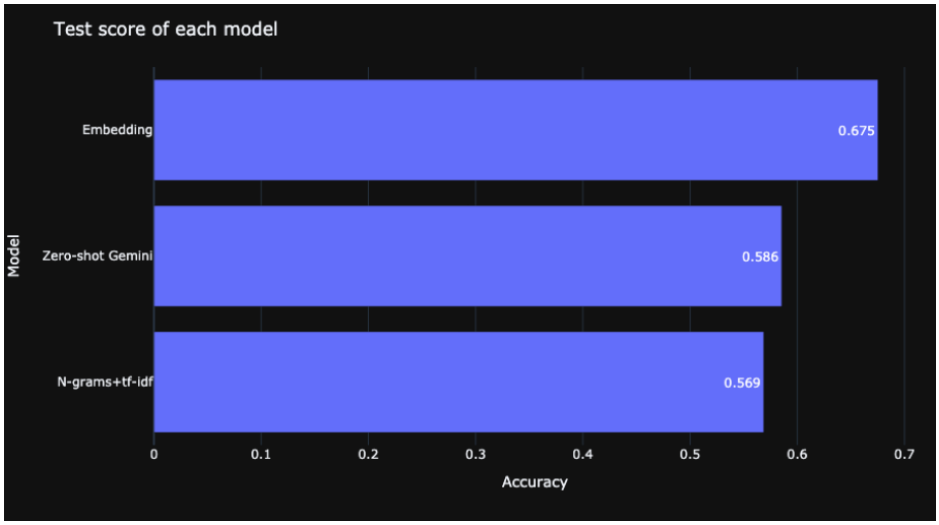


Figure 5: Test data accuracy of N-gram, zero-shot LLM, and embedding models

Summary

Overall, embedding texts before classification with XGBoost provide the higest accuracy by far. While, all models can detect clear positive and negative very well, no model excels at detect neutral, slightly positive or negative tones.