Analyzing the Effects of N-gram Models and Vocabulary Size to the Accuracy of machine learning model for Customer Sentiment Classification in Amazon Platform

Nguyễn Thiện Trí[1,2,3], Đoàn Bảo Long[1,2,3], and Đỗ Trọng Hợp[1,2,4]

[1] University of Information Technology, Ho Chi Minh City, Vietnam
[2] Vietnam National University Ho Chi Minh City, Vietnam
[3] {21522707,21520332}@gm.uit.edu.vn
[4] Correspondence: hopdt@uit.edu.vn

Abstract. Customer sentiment analysis is a process of analyzing and understanding the emotions, opinions, and attitudes expressed by customers towards a product, service, brand, or company. In this study, we are going to evaluate the impact of two factors: N-gram models and the Vocabulary size to the accuracy of a AMAZON Customer Sentiment Classification. Experiment results were analyzed using Two-way ANOVA test, TurkeyHSD and an optimal regression model is found for each N-gram model in order to find the most optimal approach for the AMAZON-Customer-Sentiment Classification. Amazon Product Reviews dataset is used to build the model. The analysis results show that there are differences between levels of each factor and an interaction exists between the two factors mentioned above, we also found the optimal Vocabulary size for this sentiment analysis task, which will help reduce the training time as well as increases the performance for the machine learning model.

Keywords: Sentiment Analysis · N-gram · Hypothesis testing · Regression · ANOVA.

1 Introduction

In the dynamic landscape of e-commerce, understanding customer sentiment is pivotal for sellers to thrive in Amazon's bustling marketplace. With millions of customers leaving feedback, reviews, and comments daily, extracting meaningful insights from this vast reservoir of information can be a daunting task. To empower sellers with actionable knowledge, Amazon employs advanced Natural Language Processing (NLP) techniques, specifically Amazon Customer Sentiment Classification. This machine learning approach allows sellers to comprehend and categorize customer sentiments effectively, enabling data-driven decision-making and enhancing their business strategies.

1.1 Research Problem

We conduct an analysis to assess the influence of these two factors on the machine learning model's accuracy, aiming to identify the optimal approach for the customers sentiment classification model.

N-gram N-gram models are a type of language model used in natural language processing (NLP) and computational linguistics. An N-gram represents a sequence of N adjacent items, which can be characters, words, or even larger linguistic units like phrases or sentences. In the context of sentiment analysis, N-gram models are commonly used to analyze and understand the sentiment expressed in text. For example, let's consider the sentence: "I really enjoyed the movie."

- A unigram (1-gram) model would consider each word in isolation: ["I", "really", "enjoyed", "the", "movie"].
- A bigram (2-gram) model would consider pairs of adjacent words: ["I really", "really enjoyed", "enjoyed the", "the movie"].
- A trigram (3-gram) model would consider triplets of adjacent words: ["I really enjoyed", "really enjoyed the", "enjoyed the movie"].

N-gram models provide a way to capture the contextual information and relationships between adjacent items in a sequence. By analyzing these patterns, the model can understand the sentiment expressed in a sentence more effectively.

Vocabulary size The size of the Vocabulary refers to the total number of unique words or tokens present in a given corpus or dataset. In the context of sentiment analysis, the Vocabulary represents the set of words that the model has been exposed to during training.

The size of the Vocabulary can vary greatly depending on factors such as the size of the dataset, the domain or topic of the text data, and the preprocessing techniques applied. In general, larger datasets tend to have larger vocabularies due to the presence of a wider range of words.

Different vocabulary sizes enable sentiment analysis models to understand the context in which words are used. By incorporating a larger vocabulary, the models can better distinguish between different senses or meanings of words, reducing ambiguity and improving the accuracy of sentiment classification.

It's important to note that while a larger vocabulary can bring benefits, it also increases the computational complexity and resource requirements for training and inference. That's why we choose to study the effect of different Vocabulary sizes on this sentiment analysis task.

1.2 Related Works

There have been several studies on the influence of N-gram models on the accuracy of machine learning models in sentiment analysis tasks [1, 2]. However, they

only use 1 level of the N-gram models, which is Bigram, meanwhile, our research analyzes the influence for 3 levels. In addition, there hasn't been any article that considers the Vocabulary size factor as well as the interaction between the two factors above. Therefore, we decided to conduct this research.

2 Experiment Setup

2.1 The Amazon Product Reviews Dataset

Our experiment dataset includes 1 million English reviews of customers about the products on Amazon and their ratings (from 1 to 5), in this study, we selected only reviews with a rating of 1, 3 and 5 (Negative, Neutral, Positive) in order to increase the consistency for the dataset's labels. Details of the Amazon Product Reviews dataset can be found at `https://nijianmo.github.io/amazon/index.html`

2.2 Text Preprocessing

After reviewing some features of the dataset, we have identified the following preprocessing methods:

- LowerCase
- Replace URL (replace all links contained in the text by "URL")
- Replace consecutive characters
- Remove short words
- Expand contractions
- Remove non-alphabet characters
- Remove stop words
- Lemmatize text

2.3 Feature Extraction

The Feature Extractor used in the experiment is TfidfVectorizer, it is also a class in the scikit-learn library, which is a popular machine learning library in Python. It is used for text feature extraction and vectorization based on the TF-IDF (Term Frequency-Inverse Document Frequency) representation.

N-gram models In the context of scikit-learn's TfidfVectorizer, the *ngram_range* parameter allows us to specify the range of n-grams to consider during vectorization. It takes a tuple of two values (min_n, max_n) that represent the minimum and maximum values of n-grams to consider. In this study, we are going to analyze the effect of 3 levels in N-gram models: Unigram, Bigram and Unigram combine with Bigram.

Vocabulary size The vocabulary size and the *max_features* parameter in TfidfVectorizer are related to the number of unique words (or features) considered for feature extraction. The *max_features* parameter is an optional parameter that limits the vocabulary size to a specified number of most frequent words. It allows us to control the size of the vocabulary by selecting only the most important or informative words. If set to an integer value, the TfidfVectorizer will consider only the *max_features* most frequent words and ignore the rest. This can be useful when dealing with large vocabularies to reduce memory usage and focus on the most significant terms. In this study, we are going to analyze the effects of 7 levels in the Vocabulary size: 10000, 20000, 30000,..., 70000.

2.4 Replication and Randomization

In order to analyze the interaction between these two factors correctly, we need to have several replications for each treatment. In statistics, replication is the repetition of an experimental condition so that the variability associated with the phenomenon can be estimated.

K-fold cross-validation In this study, we use the K-fold cross-validation technique to create 4 replications for each treatment. K-fold cross-validation is a widely used technique in machine learning and statistics to assess the performance and generalization of a predictive model. It involves partitioning the dataset into K subsets (or folds) of approximately equal size. The model is trained and evaluated K times, each time using a different fold as the validation set and the remaining K-1 folds as the training set.

After each K-fold split operation, our data is shuffled to ensure the randomization principle.

2.5 Machine Learning model

The machine learning model used in the experiment is Softmax Regression. Softmax Regression, also known as Multinomial Logistic Regression, is a widely used technique in sentiment analysis due to its effectiveness in handling multiclass classification problems. Here are some benefits of using Softmax Regression:

- Simplicity and efficiency: Softmax Regression is computationally efficient and relatively simple to implement compared to more complex algorithms like deep learning models. It requires minimal hyperparameter tuning, making it a practical choice for smaller datasets and quick prototyping.
- Low data requirements: While deep learning models often require large amounts of labeled data for training, Softmax Regression can perform reasonably well even with smaller datasets. This can be advantageous when labeled data is limited or expensive to obtain.

3 Experiment result Analysis

The machine learning model is evaluated based on its Accuracy score since our experiment dataset has an equal number of observations for each class.

3.1 Two-way ANOVA test

A two-way analysis of variance (ANOVA) is a statistical test used to analyze the effects of two categorical independent variables on a continuous dependent variable. It allows us to examine the main effects of each independent variable as well as their interaction effect.

Our two-way ANOVA test will test the following three null hypotheses:

- H0: mean accuracy of all levels of N-gram factor are equal.
- H0: mean accuracy of all levels of Vocabulary size (Number of features) factor are equal.
- H0: there is no interaction between two factors N-gram and Vocabulary size.

The results of the Two way ANOVA test are presented in the table below:

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Ngram	2	210.93	105.46	6733.62	<2e-16	***
Number_of_Features	6	23.41	3.90	249.16	<2e-16	***
Ngram:Number_of_Features	12	17.94	1.50	95.47	<2e-16	***
Residuals	84	1.32	0.02			

Fig. 1. Two way ANOVA test table

From the figure above, we can see that all 3 p-values are less than 0.05, so we can reject the null hypothesis H0 at the significance level of 5%, which means there are differences between the mean accuracy of levels of each factor and there is an interaction between N-gram models and Vocabulary size.

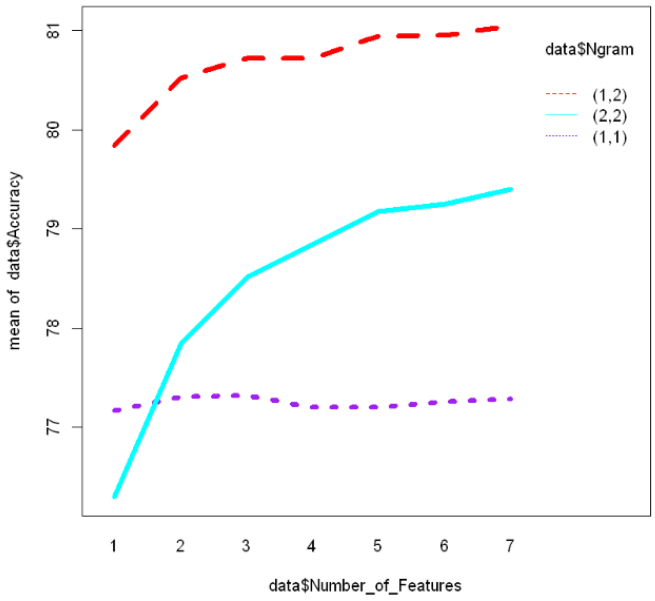

Fig. 2. Interaction plot

In the figure above, we use MinMaxScaler to scale values on the x-axis from (10000, 70000) to (1, 7), this makes it easier to observe the coefficients of the regression model. And N-gram = (1,1) means Unigram only, (2,2) means Bigram only, (1,2) means Unigram combine with Bigram. From the interaction plot, we can clearly see the interaction between N-gram models and Vocabulary size, especially at N-gram = (2,2), while the accuracy of other N-gram models doesn't increase significantly when the Vocabulary size increases, the accuracy of the N-gram (2,2) model increases quite markedly.

```
$Ngram
                diff       lwr       upr p adj
(1,2)-(1,1)  3.425297  3.353917  3.496676     0
(2,2)-(1,1)  1.222451  1.151071  1.293830     0
(2,2)-(1,2) -2.202846 -2.274225 -2.131467     0
```

Fig. 3. TurkeyHSD test

The plot and TurkeyHSD test also show that the N-gram = (1,2) model gives the highest results among the three models. There might be a few reasons why N-gram = (1,2) performs better than the others:

– Capturing more context: By including both unigrams and bigrams, an (1,2) model can capture more contextual information. Unigrams provide infor-

mation about individual words, while bigrams capture some degree of word ordering and dependencies. This expanded context can be beneficial for tasks that require a better understanding of language structure and semantics.
– Handling negations and modifiers: Negations and modifiers play an important role in sentiment analysis. Negations like "not good" can reverse the sentiment of a phrase, and modifiers like "very good" can intensify it. By considering both unigrams and bigrams, the (1,2) model can effectively capture these linguistic patterns and correctly interpret the sentiment.

3.2 Regression

– For N-gram = (1,1)

```
Call:
lm(formula = y ~ x)

Residuals:
      Min        1Q    Median        3Q       Max
-0.286620 -0.079235 -0.000972  0.061939  0.278575

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)
(Intercept) 77.231201   0.048559 1590.471   <2e-16 ***
x            0.005201   0.010858    0.479    0.635
```

Fig. 4. Linear Regression model

From the figure above, the p-value of the x coefficient= 0.635 > 0.05 shows that the results are not linearly correlated with x (Vocabulary size) at the significance level of 5%, we can also see that clearly in the plot.
– For N-gram = (2,2)

```
Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-0.44772 -0.13931  0.05346  0.12592  0.38194

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 80.00326    0.08393 953.265  < 2e-16 ***
x            0.16851    0.01877   8.979 2.23e-10 ***
```

Fig. 5. Linear Regression model

```
pureErrorAnova(Linear_Reg)
```

A anova: 4 × 5

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
x	1	29.2542885	29.25428853	2394.3765	1.142008e-28
Residuals	33	7.2695537	0.22028951	NA	NA
Lack of fit	5	6.9274521	1.38549042	113.3983	1.093773e-17
Pure Error	28	0.3421016	0.01221791	NA	NA

Fig. 6. Lack of fit test

From Fig.4, the x-coefficients of the model are statistically significant, but the p-value in lack of fit test is less than 0.05 showing that the model lacks of fit at the significance level of 5%. In a regression analysis, the lack of fit test evaluates whether the model adequately captures the systematic variation in the data. It specifically tests whether the residual variation, which represents the unexplained variation in the data, is significantly different from the pure error variation, which represents random fluctuations or noise. The lack of fit also occurs in the Quadratic Regression and Cubic Regression models. The optimal regression model for N-gram = (2,2) is Quartic Regression:

```
pureErrorAnova(Quartic_Reg)
```

A anova: 7 × 5

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
x	1	29.25428853	29.25428853	2394.376503	1.142008e-28
x2	1	5.89192666	5.89192666	482.236673	3.445976e-19
x3	1	0.88369610	0.88369610	72.327897	3.022770e-09
x4	1	0.10244328	0.10244328	8.384678	7.260024e-03
Residuals	30	0.39148768	0.01304959	NA	NA
Lack of fit	2	0.04938606	0.02469303	2.021051	1.513974e-01
Pure Error	28	0.34210162	0.01221791	NA	NA

Fig. 7. Quadrtic Regression - Lack of fit test

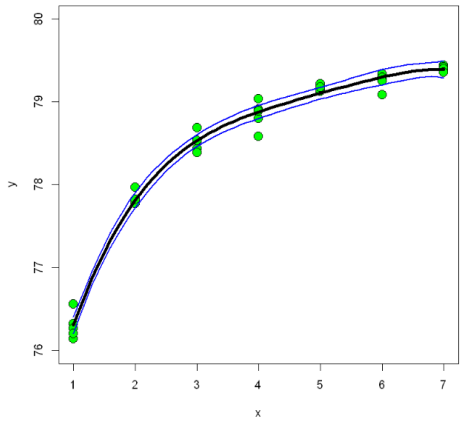

Fig. 8. Quartic Regression with 95% confidence interval

– For N-gram = (1,2), Doing the same process as above, the optimal regression model for N-gram (1,2) is a Cubic Regression:

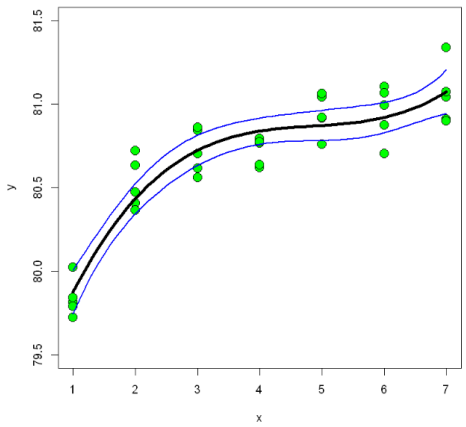

Fig. 9. Cubic Regression with 95% confidence interval

4 Future Works

Currently, the model has only been built using one dataset, but in the future, we may have the opportunity to incorporate multiple datasets, allowing us to factor in the dataset variability during experiments. Additionally, we can introduce various preprocessing steps, as well as various machine learning models into the experimentation process to explore a more suitable and effective approach for building a better sentiment classification model.

5 Conclusion

During the experiment, we observed differences in accuracy between levels for each factor (N-gram models and Vocabulary size) and their interaction.

Specifically, we found that the Ngram model combining unigram and bigram with a Vocabulary size of 70,000 achieved the highest accuracy among the tested models. Actually, accuracy always increases when Vocabulary size increases. However, from the plot and Regression models, we discovered that increasing the Vocabulary size beyond 60,000 did not lead to a significant improvement in accuracy, but it will require a lot of computational resources as well as time in order to train the machine learning model (this dataset has an original Vocabulary size of more than 400,000 words, even after going through text preprocessing). As a result, the optimal Vocabulary size appeared to be around 60,000 features.

Based on these findings, we recommend utilizing the Ngram model that combines unigram and bigram with a Vocabulary size of approximately 60,000 features for future sentiment classification tasks, as it demonstrated a suitable and efficient approach to the task at hand.

References

1. Ravinder Ahuja, Aakarsha Chug, Shruti Kohli, Shaurya Gupta, Pratyush Ahuja, The Impact of Features Extraction on the Sentiment Analysis, Procedia Computer Science, Volume 152, 2019, Pages 341-348, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2019.05.008
2. Hung, L.P., Alfred, R. (2017). A Performance Comparison of Feature Extraction Methods for Sentiment Analysis. In: Król, D., Nguyen, N., Shirai, K. (eds) Advanced Topics in Intelligent Information and Database Systems. ACIIDS 2017. Studies in Computational Intelligence, vol 710. Springer, Cham, https://doi.org/10.1007/978-3-319-56660-3_33
3. Justifying recommendations using distantly-labeled reviews and fined-grained aspects Jianmo Ni, Jiacheng Li, Julian McAuley Empirical Methods in Natural Language Processing (EMNLP), 2019