Naïve bayes and maximum entropy comparison for translated novel’s genre classification

M A Aditya, A Helen*, and I Suryana

Department of Computer Science, Faculty of Mathematics and Natural Science, Universitas Padjadjaran, Jl Raya Bandung Sumedang KM 21 Jatinangor Sumedang 45363, Indonesia

*helen@unpad.ac.id

Abstract. In the last two decades, novel translation had become one of the popular products among the literature community. People had favorited some genre based on their ages. The reader needs to finish reading until the end first before they could determine what genre a novel should have. There were some cases where the genre written in the description differs from the real novel's content, which made readers felt upset and had not pleasant reading experience. This research is going to do classification for the novel's genre automatically. Naïve Bayes is the method chosen for classification, later the result of Naïve Bayes classification is going to be compared with another algorithm, which is Maximum Entropy algorithm. Each method would apply algorithms to label the data based on an existing class. The data origin was taken from 12 translated novel that has 3746 lines. Data was portioned into three genre classes, "Action-Fantasy" for about 1293 lines, "Modern-Slice-of-Life" for 1203 lines, and 1250 for "Other". Evaluation of the two models, Naïve Bayes and MaxEnt, would be using confusion matrix that generated the highest number precision, recall, and F-score which values were 77,52%; 75,59%; and 77,55% for the Naïve Bayes method, and 78,11%; 83,82%; and 75,81% for MaxEnt method. The value of accuracy were 72,72% for Naïve Bayes, and 71,86% for MaxEnt. Both methods showed that the genre "Action-Fantasy" was the correct genre for almost every novel among 12 novels listed.

1. Introduction

Ever since industrial revolution happened, print media became far easier to produce thanks to the printing press, which also mean easier to get by various kind of people, and at the same time, literature culture also growth along. The story printed, which is called Novel, became one of the entertainments consumed by many people.

With passage of time, some circles of people have read a novel as their hobby. But, some things prevent people to fully enjoy the content of the novel, one of them is language. Language becomes one of the biggest problems because the language novel written to is affected by from where the writer is. The problem of language difference is then solved by the emerging of Novel Translator.

In the last two decades, translated novel had become a popular product for avid readers. The reason is that quality and uniqueness of the novel's content written in Japanese, Chinese, and Korea, which have non-Latin alphabets, are in great demand and also translated into globally accepted language, which is English.

Novel has many kinds of characteristics like theme, backstory, character, and writing style which everything has their own fans. When more than one characteristic becomes related to another, that relationship is then called by genre.
Genre represents the content of the novel, so the reader has an image of what kind of novel it is other than the summary provided by the publisher. But there were some cases where a genre written in the description of the novel is not accurate or even wrong altogether, which makes more sense if another genre is written. This problem could make the reader felt upset and had not pleasant reading experience, which leads to the downfall of the novel's reputation itself.

Seeing that, we have a suggestion to make a help system which in the form of analysis whether a genre written on the novel is correct or not, to not failing readers when they choose to read a novel based on their favorite genre. Because of this problem, I proposed to make an automatic novel's genre classification, which the result could be compared with genre manually assigned by the novel's reader and translator. There are two classification methods used in this research, they are Naïve Bayes and Maximum Entropy(MaxEnt), or also could be called Logistic Regression.

2. Method
2.1 Data Origin
The data origin was taken from 12 translated novel that has a total of 3746 lines. Data was portioned into three genre class, as mentioned in Table 1.

| Genre               | Lines |
|---------------------|-------|
| Action-Fantasy      | 1293  |
| Modern-SoL          | 1203  |
| Other               | 1250  |

2.2 Text Preprocessing
Novel translation described as substitution of the novel's native language text into text has equal meaning in destination language which also takes into consideration other aspects like a joke, saying, and also many cultural related terms. In Japanese to English novel translation, other than increase word count caused by translation, there is also addition of words like "are", "were", and "then". These words could create a problem because of non-consistent variable input into the corpus used for genre classification. To prevent this, we need to do a text preparation called by Text Preprocessing

Text Preprocessing is one of the big factors to determine the performance of a model. Noise removal, case folding, stop word removal, tokenization, and stemming is the usual process of Text Preprocessing [1]. The main use of doing Text Preprocessing is to increase the accuracy of information extraction.

Dataset was going to be cleaned by various sub-steps in Text Preprocessing. The first step is Noise removal, which deletion of number and special character from data, continued with Case folding, where is a process to turn every character in data into lower case. Then there is Tokenization which separated every word inside data to turn data become an array of words instead of a line, lastly, Stopword removal would remove unnecessary words inside data.

2.3 Term-Frequency Inverse-Document-Frequency
The classification Algorithm can't process data as original text format, they needed to be converted into vector feature form first. The feature is an entity without internal structure which will become a dimension inside vector space. Each feature has a weight that represents the document [2]

Term-Frequency Inverse-Document-Frequency (TF-IDF) is one of the popular weight schemas for a document to vector conversion, Term-Frequency (TF) is frequency of how many times a term shows up in a document. Document-Frequency (DF) is the number of documents which has a certain term. The higher the number of frequencies a term has, also higher the weight it has. But, a term with high weight doesn't always mean it has an important role in a document.
To reduce the domination of that kind of term, there is something called Inverse Document Frequency (IDF). A term with low frequency will have a high IDF score, vice versa term with high frequency will have a low IDF score. After we obtained TF and DF score, there will be TF-IDF calculation to generate a composite weight score from every term in every document. The calculations are displayed as equation (1) and (2). [3]

\[
\text{idf}_t = \log \frac{N}{df_t} \tag{1}
\]

\[
\text{tf}_t \cdot \text{idf}_{t,d} = \text{tf}_{i,d} \cdot \text{idf}_t. \tag{2}
\]

In this research, data after Text Preprocessing will be converted into a vector using TfIdfVectorizer class from NLTK library.

2.4 Naïve Bayes

Naïve Bayes is one of the generally used classification methods to solved classification problems. This method is based on Bayes theorem with independent and naïve assumption. This method has high accuracy score even with a low number of training data. Besides, Naïve Bayes is very efficient at computation process either from CPU usage or Memory usage. [4]

Based on Bayes theorem, the probability from a hypothesis based on evidence is equal to probability of that hypothesis multiple by probability from evidence which gave the hypothesis, then the result is divided by probability from evidence. From that, we could conclude Bayes theorem as equation (3).

\[
P(H|E) = \frac{P(H) \cdot P(E|H)}{P(E)}. \tag{3}
\]

Bayes theorem gave us a mean to find the probability of hypothesis based on evidence. But at classification problem, there are more than one evidence, and every evidence is assumed as independent, so we got a Naïve assumption which turns equation (3) into (5).

\[
P(H|E_1, E_2, ..., E_n) = \frac{P(H) \cdot P(E_1|H) \cdot P(E_2|H) \cdot ... \cdot P(E_n|H)}{P(E_1) \cdot P(E_2) \cdot ... \cdot P(E_n)}. \tag{4}
\]

Equation (4) could be simplified as (5)

\[
P(H|E_1, E_2, ..., E_n) = \frac{P(H) \cdot \prod_{i=1}^{n} P(E_i|H)}{P(E_1, E_2, ..., E_n)}. \tag{5}
\]

If we see this from the Text Classification viewpoint, probability is the prediction score, evidence is token which consists of words of data that are going to be classified, and hypothesis is class that going to be used for classification, which means genres. So equation (5) could also be written as equation (6).

\[
P(c|t_1, t_2, ..., t_n) = \frac{P(c) \cdot \prod_{i=1}^{n} P(t_i|c)}{P(t_1 \cdot t_2 \cdot ... \cdot t_n)}. \tag{6}
\]

To applied method in the program, equation (6) must be turned into a model first. With \(P(E_1, E_2, ..., E_n)\) is constant depended on the input given, we got a model displayed as equation (7).

\[
P(c|t_1, t_2, ..., t_n) \propto P(c) \cdot \prod_{i=1}^{n} P(t_i|c)
\]

↓↓↓
\[ \hat{c} = \arg \max_c P(c) \cdot \prod_{i=1}^{n} P(t_i|c). \]  

(7)

After that, the model is ready to be used in the program. The flow of data testing is displayed in Figure 1.

![Flowchart](image)

**Figure 1.** The flow of naïve bayes data testing

### 2.5 Maximum Entropy

Maximum Entropy (MaxEnt) classification, or also could be called Logistic Regression, is one of the generally used Machine Learning methods to solve Text Mining problems like Sentence Boundary Detection, text categorization, Machine Translation, and many other. [5]

This classification method has an advantage in data classification with a non-independent assumption, which is often found in the Text Classification problem. But at the same time, MaxEnt has a glaring problem around the very long time needed to train data compared to Naïve Bayes, this is because optimization problem at solving a calculation needed by the model to obtain an estimation parameter.

To use MaxEnt for classification, first is to construct a stochastic model [6], which accurately represents the behavior of the random process: take as input the contextual information evidence of a document and produce the output value hypothesis. The second step is to summarize the training sample in terms of its empirical probability distribution as displayed in equation (8).

\[ \bar{P}(E,H) = \frac{1}{N} \cdot \text{number of times } (E,H) \text{ occurs in the sample.} \]  

(8)

Using equation (8), empirical probability distribution to construct the statistical model of the random process which assigns texts to a particular class by taking into account their contextual information. The building blocks of the model will be the set of statistics that come from the training dataset. Indicator function is displayed as equation (9).

\[ f_j(E,H) = \begin{cases} 1 & \text{if } H = c_i \text{ and } E \text{ contains } w_k \\ 0 & \text{else} \end{cases}. \]  

(9)

The indicator function at equation (9) could also be called "feature". This binary-valued indicator function returns 1 only when the class of a particular document is \( c_i \) and the document contains the word \( w_k \). Then, by expressing any statistic of the training dataset as the expected value of the appropriate binary-valued indicator function \( f_j \). The expected value of feature \( f_j \) with respect to the empirical distribution is equal to equation (10).

\[ \bar{P}(f_j) \equiv \sum_{E,H} \bar{P}(E,H)f_j(E,H). \]  

(10)

If each training sample \((E,H)\) occurs once in training dataset then, \( \bar{P}(E,H) \) is equal to \(1/N\). Next step is to create a model according to equation (10). With constraining the expected value that the model assigns to the expected value of the feature function \( f_j \). The expected value of feature \( f_j \) with respect to the model \( P(E,H) \) is equal to equation (11).
\[ P(f_j) = \sum_{E,H} \tilde{P}(E)P(H|E)f_j(E,H). \] (11)

Where \( \tilde{P}(E) \) is the empirical distribution of evidence in the training dataset and it is usually set equal to \( 1/N \). By constraining the expected value to be equal to the empirical value and from equations (9) and (10) we got equation (12).

\[ \sum_{E,H} \tilde{P}(E)P(H|E)f_j(E,H) = \sum_{E,H} \tilde{P}(E,H)f_j(E,H). \] (12)

Equation (12) is called constrain and could be created as many constrains as the number of \( f \) feature functions. According to the principle of MaxEnt, the model which as close as possible to equation (11) is \( P^* \) model, as displayed in equation (13) below.

\[ P^* = \arg \max_{P=\mathcal{E}} \left( -\sum_{E,H} \tilde{P}(E)P(H|E) \log P(H|E) \right) \] (13)

where :
1. \( P(H|E) \geq 0 \) for all \( E,H \)
2. \( \sum_H P(H|E) = 1 \) for all \( E \)
3. \( \sum_{E,H} \tilde{P}(E,H)f_j(E,H) = \sum_{E,H} \tilde{P}(E,H)f_j(E,H) \) for \( j \in \{1,2,...,n\} \).

Next step is to solve the optimization problem using the Lagrangian multipliers, with focusing on the unconstrained dual problem and give an estimation to the lamda free variables \( \{\lambda_1,...,\lambda_n\} \) with the Maximum Likelihood Estimation method. After that, the probability given an evidence to be classified as a hypothesis is equal equation (14).

\[ P^*(H|E) = \frac{\exp(\sum_i \lambda_i f_i(E,H))}{\sum_H \exp(\sum_i \lambda_i f_i(E,H))}. \] (14)

As long as lamda parameters for the model are found, next step is to use Maximum a posteriori (MAP) decision rule and select the category with the highest probability. For estimating the lamda parameters required using an iterative scaling algorithm such as the GIS (Generalized Iterative Scaling) or the IIS (Improved Iterative Scaling).

Other than searching for estimation of lamda parameter, feature function \( f^*(E,H) \) which the total number of features which are active for a particular \( (E,H) \) pair also needed to be found. If this number is constant for all documents then the \( \Delta \lambda_i \) can be calculated as equation (15).

\[ \Delta \lambda_i = \frac{1}{C} \log \frac{\tilde{P}(f_j)}{P(f_j)} \text{ where } C = f^*(E,H). \] (15)

The assumption that \( f^*(E,H) \) is constant is rarely realistic in practice, but this won’t be a problem as long as the algorithm could generate result as long as the sum of indicator functions to be bounded by \( f^*(E,H) \) and not necessarily equal to it. Then, assign \( C \) as the maximum number of active features for all \( (E,H) \) pairs within training dataset, we got equation (16) to find the parameter \( \{\lambda_1,...,\lambda_n\} \) as the model.

\[ C = C_{\text{max}} = \arg \max_{E,H} f^*(E,H). \] (16)

Along with the model, the flow of data testing is displayed as Figure 2.
Confusion Matrix, or also called by error matrix is a kind of table used to display the performance of a classification model. In this research, model based on both methods will be evaluated with the confusion matrix which contains the prediction result. There are four elements of the confusion matrix, a positive class predicted as positive (true positive/Tp), a negative class predicted as negative (true negative/Tn), a positive class predicted as negative (false positive/Fp), and a negative class predicted as positive (false negative/Fn) [7].

Matrix must have a size of $m \times m$, where $m$ is the sum of class. Confusion matrix for two-class classification is displayed in Table 2 below.

| Actual  | Predicted |          |          |
|---------|-----------|----------|----------|
| Positive| Tp        | Fp       |          |
| Negative| Fn        | Tn       |          |

Confusion Matrix could be used to calculate accuracy, recall, precision, dan $F$-score from true positive (Tp), true negative (Tn), false positive (Fp), dan false negative (Fn) score [8]. Accuracy is the percentage of correctly predicted data, the equation (17) show how to calculate it.

$$\text{Accuracy} = \frac{Tp + Tn}{Tp + Tn + Fp + Fn} \quad (17)$$

Recall is the percentage of correctly predicted positive data, the equation (18) show how to calculate it.

$$\text{Recall} = \frac{Tp}{Tp + Fn} \quad (18)$$

Precision is the percentage of positive data predicted as positive, the equation (19) show how to calculate it.

$$\text{Precision} = \frac{Tp + Tn}{Tp + Fp} \quad (19)$$

$F$-Score is a representation of recall and precision, the equation (20) show how to calculate it.

$$F - \text{score} = \frac{2 * \text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})} \quad (20)$$

2.6 Confusion Matrix

2.7 Genre
Genre is a term that is often mentioned in modern society. Genre is a vague concept without a clear boundary, and no need to be united [9]. Also according to The American heritage dictionary of the English language, genre is perceived as a category of artistic composition, as in music or literature, marked by a distinctive style, form, or content.

From two definition of genre above, we could conclude that genre is a component influence a creation different from the other, but along with emerging another creation which has similar component, the meaning of genre change from just owned by the first creation has that component, into a symbol of any creation who has similar component.

3. Result and Discussion
Implementation of both Naïve Bayes and MaxEnt is using the sklearn library[10]. Data post TF-IDF was then divided into 10-fold first using StratifiedKFold before predicted using the model. The prediction result of both methods could be seen in Table 3 below.

| Table 3. Result of testing data by stratified 10-fold. |
|-----------------|-----------------|-----------------|
| Naïve Bayes | MaxEnt |
| Average Accuracy of 10-fold | 73% | 72% |
| Train duration | 12.366 s | 296.057 s |
| Test duration | 1.311 s | 1.294 s |

From Table 3, we could see while the average accuracy and test duration of both methods didn’t show a big difference, Naïve Bayes has significantly lower train time instead of MaxEnt. This had been mentioned before as MaxEnt has the optimization problem at solving a calculation needed by the model. Aside from that, there are also Confusion matrix showing the predicted result for both methods displayed as Tables 4 and 5 below.

| Table 4. Confusion matrix for naïve bayes method |
|-----------------|-----------------|-----------------|
| Actual Genre | Predicted Genre |
| Action-Fantasy | Modern-SoL | Other |
| Action-Fantasy | 948 | 175 | 170 |
| Modern-SoL | 253 | 807 | 143 |
| Other | 185 | 96 | 969 |

| Table 5. Confusion Matrix for MaxEnt Method |
|-----------------|-----------------|-----------------|
| Actual Genre | Predicted Genre |
| Action-Fantasy | Modern-SoL | Other |
| Action-Fantasy | 1010 | 199 | 84 |
| Modern-SoL | 303 | 817 | 83 |
| Other | 244 | 141 | 865 |

From two confusion matrix tables, we could summaries the predicted result according to each genre as Table 6 below.

| Table 6. Predicted result based on class |
Then, we could generate the evaluation for both methods by calculating the Precision, Recall, F-Score and Accuracy displayed in Tables 7 and 8 below.

**Table 7. Evaluation of naïve bayes model**

| Genre            | Action-Fantasy | Modern-SoL | Other |
|------------------|----------------|------------|-------|
| Precision        | 73.32%         | 67.08%     | 77.52%|
| Recall           | 68.40%         | 74.86%     | 75.59%|
| F-Score          | 70.77%         | 70.76%     | 77.55%|
| Accuracy         |                |            | 72.72%|

**Table 8. Evaluation of MaxEnt model**

| Genre            | Action-Fantasy | Modern-SoL | Other |
|------------------|----------------|------------|-------|
| Precision        | 78.11%         | 67.91%     | 69.20%|
| Recall           | 64.87%         | 70.61%     | 83.82%|
| F-Score          | 70.88%         | 69.23%     | 75.81%|
| Accuracy         |                |            | 71.86%|

4. Conclusion

Evaluation of the two models, Naïve Bayes and MaxEnt, would be using confusion matrix that generated the highest number precision, recall, and F-score which values were 77.52%; 75.59%; and 77.55% for the Naïve Bayes method, and 78.11%; 83.82%; and 75.81% for MaxEnt method. The value of accuracy were 72.72% for Naïve Bayes, and 71.86% for MaxEnt.

Either Naïve Bayes and Maximum Entropy as classification method could be used to predict the translated novel's genre, as aside from train time duration, both methods perform similarly by showing not too different in various evaluation scores ranging from Precision, Recall, F-Score, and especially accuracy. If we looked from number alone, Naïve Bayes has an advantage for about 0.86% better than Maximum Entropy, but with Maximum Entropy has an attribute of non-independent, we could disregard the little difference and more believe result generated by Maximum Entropy classification model.

Between three classes, Action-Fantasy was the correct genre for almost every novel among 12 novels listed. Although the train data could be improved more as currently it only shows about 72% accuracy even with the model optimized for text classification. Also, there was a suggestion to recreate the same research but using novel with the Indonesian language.

Acknowledgments
The authors thank to Rector Universitas Padjadjaran who provided financial assistance for the dissemination of research through Academic Leadership Grant with contract number: 1427/UN6.3.1/LT/2020. Also thanks to the reviewers for the valuable review of this paper.

References

[1] Uysal, Kürşat A, and Gunal S 2012 A novel probabilistic feature selection method for text classification Knowledge-Based Systems 36, 226-235. 10.1016/ j.knosys. 2012.06.005.
[2] Manning C D, Raghavan P, and Schütze H 2008 Introduction to Information Retrieval (Cambridge: Cambridge University Press)
[3] Liu B, 2012 Sentiment Analysis and Opinion Mining Synthesis Lectures on Human Language Technologies
[4] Nigam K, Lafferty J D, and McCallum A 1999 Using Maximum Entropy for Text Classification
[5] Cuong N N , Linh N T , Thụy H Q, and Hieu P X 2007 A Maximum Entropy Model for Text Classification
[6] Berger A L, Pietra V J D, and Pietra S A D 1996 A maximum entropy approach to natural language processing Computational Linguistics 22 1 (March 1996) 39–71.
[7] Sokolova M, and Lapalme G 2009 A systematic analysis of performance measures for classification tasks Information Processing & Management 45 427-437. 10.1016/j.ipm.2009.03.002
[8] Patro V M, and Patra M R 2015 A Novel Approach to Compute Confusion Matrix for Classification of n-Class Attributes with Feature Selection Transactions on Machine Learning and Artificial Intelligence 3(2) 52 https://doi.org/10.14738/tmlai.32.1108
[9] Finn A, and Kushmerick N 2006 Learning to classify documents according to genre Journal of the American Society for Information Science and Technology 57 1506-1518 doi: 10.1002/asi.20427
[10] J. Huang, J. Lu and C. X. Ling 2003 Comparing naive Bayes, decision trees, and SVM with AUC and accuracy Third IEEE International Conference on Data Mining Melbourne, FL, USA, pp. 553-556 doi: 10.1109/ICDM.2003.1250975