|  |
| --- |
| The 28th Meeting of the Wiesbaden Group on Business Registers |
| Name of Author  Annisarahmi Nur Aini Aldania, Irma Damayanti  Organization  BPS – Statistiscs Indonesia  Contact address  BPS – Statistics Indonesia  Directorate of Census and Survey Methodology Development  Jl. Dr. Sutomo No. 6-8 Jakarta 10710 Indonesia  [annisarahmi@bps.go.id](mailto:annisarahmi@bps.go.id), [irma.damayanti@bps.go.id](mailto:irma.damayanti@bps.go.id)  Contact phone  +62 8118901610 (Aldania)  +62 8194887771 (Damayanti) |
| Title of Paper  Artificial Intelligence for Predicting Indonesia Industrial Classification Code |

Abstract

Industrial classification is a rule or principle for grouping establishments based on their economic activities into specific classes. BPS - Statistics Indonesia uses the Indonesia Industrial Classification Code (KBLI) as a classification code guideline for economic activities. In the KBLI guidelines, each establishment/enterprise can be classified into five-digit KBLI subclasses according to their main activities and products. Artificial intelligence can be used to automate industrial classification. This study compares the performance of Double Random Forest and Fine-tuning IndoBERT in carrying out multi-class classification for KBLI codes. Two sections (G and I), which include 16 KBLI subclasses, will be predicted. We are using two scenarios to build a model: first, we build one model to predict all KBLI without differentiating their section, and second, we build two models for each section. This research shows that fine-tuning IndoBERT can provide higher accuracy than DRF for both scenarios. IndoBERT can achieve 96% accuracy in the first scenario, while DRF achieves 95%. IndoBERT achieves 98% accuracy in the second scenario, while DRF achieves 96% accuracy. These results also show that the model built for each section gives better results than one model for all sections.

1. Introduction

Industrial classification is a rule or principle for classifying economic activities into certain classes or categories. Each establishment can be grouped into industrial classifications based on their similarities in activities and products. In Indonesia, BPS - Statistics Indonesia adopted the International Standard Industrial Classification of All Economic Activities (ISIC) to form the Indonesia Industrial Classification Code (KBLI).

The KBLI is arranged hierarchically: one-digit numbers for sections, two-digit numbers for division, three-digit numbers for groups, four-digit numbers for classes, and five-digit numbers for subclasses. Each establishment can be classified up to the five digits of the KBLI. There are 1789 subclasses (five-digit KBLI) based on the 2020 KBLI guidelines.

KBLI 2020 is an update of KBLI 2015, which was previously published. KBLI 2020 still refers to ISIC Rev. 4 to 4 digit level, so it has the same conceptual basis as ISIC Rev. 4. Thus, the statistical data presented according to the KBLI can be compared internationally.

To determine the KBLI code of an establishment, there are questions about "main activities" and "main products" on the BPS - Statistics Indonesia questionnaire. The main activity is an approach to knowing the economic activities. Meanwhile, the main product is an approach to knowing the products produced from the main activity. Supervisors determine the KBLI code during the census/survey based on the enumerator's results of their interviews with the establishment. After they write down the main activities and products for an establishment, the supervisors will search for the appropriate KBLI code based on these details.

KBLI is important in business-based databases or frames, including in the Statistical Business Register (SBR) database. The majority of business data on the SBR comes from the results of the 2016 economic census and other surveys using the 2015 KBLI. During its development, there were changes in the KBLI classification code. Currently, the classification code used is KBLI 2020. For this reason, it is important to map existing and future establishments into the 2020 KBLI code.

To classify each establishment into KBLI codes automatically, an artificial intelligence machine learning model can be used. A textual description of the establishment’s main activities and main products can be used as input to predict KBLI. Thus, classification models can be an alternative to manual mapping.

This research uses Double Random Forest [1] and fine-tuned IndoBERT [2] to classify establishment into 2020 KBLI subclasses. We use two variables as input from the 2016 economic census dataset: the establishment’s main activities and products. We also use the five-digit KBLI codes as a class for the classification model. This data has been mapped to KBLI 2020 manually by experts. Lastly, the model formed is evaluated using accuracy and F1 Score.

This study is organized as follows: first is a literature review. In the literature review, an overview of similar research that has been carried out will be provided. Next is the methodology; this chapter will explain the data and methods we used to build the model. Following that is the results and discussion chapter, which will explain the result from our model, and finally, the conclusion that will summarize this study.

1. Literature Review
   1. Text Classification

Text classification is a process for categorizing texts into classes. In general, the stages in text classification can be divided into pre-processing, feature extraction, dimension reduction, determining classification techniques (classifiers), evaluation, and prediction of new text data based on the selected classifier [3] [4] (Figure 1).

A diagram of a model

Description automatically generated

Figure 1 Text classification steps

* 1. Industry Classification using Machine Learning

Several studies have been carried out regarding industrial classification using machine learning and deep learning models, including:

* Wood *et al.* (2017) predicted six-digit NAICS (North American Industry Classification) classification codes using deep neural networks. The input data used is a company description with a target of 1057 classes [5].
* Slavov *et al*. (2019) used modifications of BERT, XLNet, Glove and ULMfit to classify companies into an industry classification scheme with company descriptions from DBpedia used as input. Based on this research, industrial classification automation using text data can be completed with machine learning [6].
* Dwicahyo dan Yuniarto (2020) apply deep learning models, namely Gated Recurrent Unit, fastText word vectors, and label smoothing regularization (LSR) loss to automate KBLI code. Descriptions of a company's main activities and main products are used as input, with the five-digit KBLI code as an output. The data used a sample from the 2016 economic census and the 2015 KBLI guidebook. This model can produce an accuracy of 48.24% [7].
  1. Text classification with DRF

Various methods can be used to solve classification problems, including a decision tree-based method. Decision trees work in classification by dividing data based on certain rules. The rules are formed based on the data’s features. The data will be divided until it can be grouped at a more homogeneous final node. Ensemble methods that combine several single decision trees are then developed to produce better predictions. One ensemble method that can be used is Double Random Forest (DRF), a development of random forest.

Random forest (RF) forms trees in parallel using bootstrap resampling [8]. The trees formed in RF are not correlated with each other. The final prediction in RF is based on the majority of votes from all the trees formed. Double random forest (DRF) is a development of the RF model that can improve model performance when the RF model is underfitting [1]. Some differences between DRF and RF are that DRF uses all the training data to create trees at each ensemble stage so that all trees in DRF are formed using the same data from the start. Ensembles in DRF generally consist of trees that are larger than RF. In addition, DRF also uses bootstrap resampling at each node to determine the best partition rule by randomly selecting part of the features.

* 1. Text Classification with IndoBERT

The complex architecture of deep learning means that models built using deep learning require a lot of training data and resources. One solution without building a model from scratch is to use a pre-trained language model (PLM). PLM is a language model trained on a large dataset but remains agnostic to the specific problem to be used. There are many PLM representations; one that performs well in many specific tasks is BERT [9].

BERT stands for Bidirectional Encoder Representations from Transformers. BERT is designed to train a deep bidirectional representation of unlabeled text by joint conditioning on the left and right contexts in all layers. BERT is trained using Masked Language Modeling (MLM), which randomly masks several tokens in the text and independently recovers the masked tokens by predicting the highest probability of a token in the vocabulary/dictionary. Furthermore, the pre-trained BERT model can be fine-tuned by adding one output layer to form a reliable model for various tasks [9].

IndoBERT is a BERT-based model for Bahasa Indonesia. Two versions of BERT are trained using Bahasa Indonesia, namely IndoBERT, created by IndoNLU [2] and IndoBERT, created by IndoLEM [10]. The dataset used by IndoNLU is Indo4B, collected from freely available social media texts, blogs, news and websites of around 4 billion words (23 GB). Meanwhile, the dataset used by IndoLEM comes from Wikipedia (74 million words), news articles (55 million words) and the Indonesian Web Corpus (90 million words). This research will then use IndoBERT created by IndoNLU.

IndoBERT has been used in text classification tasks with several datasets, namely EmoT (classification of feelings from Twitter), SmSA (sentiment analysis from comments and reviews from various online media in Indonesia), CASA (aspect-based sentiment analysis on car review datasets), and HoASA (aspect-based sentiment analysis on hotel review dataset). IndoBERT performs quite well on this dataset compared to other pre-trained models [2]. IndoBERT is also used to classify misinformation related to COVID-19 using Twitter data. This research shows that IndoBERT has the highest performance compared to other feature-classifier combinations, such as Naïve Bayes, SVM, and Random Forest, on relevant tweet classification [11].

1. Data and Methodology
   1. Data

This study uses establishment data from the 2016 economic census. We use the establishment's main activities, main products, and five-digit KBLI codes for building the model. This research is limited to the establishment in DKI Jakarta Province, and only two sections (G and I) with 16 five-digit KBLI code subclasses will be used. This study does not use KBLI with a frequency of less than 100. An expert will manually code the establishment into the 2020 KBLI to ensure the data used has the right KBLI.

Table 1 Section, KBLI, and Code Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Section** | **KBLI** | **Description** | **Freq** |
| G | 47111 | Retail Trade of Various Kinds of Goods, Mainly Food, Drinks or Tobacco in Minimarkets/ Supermarkets/ Hypermarkets | 215 |
|  | 47112 | Retail Trade of Various Kinds of Goods, Mainly Food, Drinks or Tobacco not in Minimarkets/ Supermarkets/ Hypermarkets (traditional) | 4077 |
|  | 47213 | Vegetable Retail Trade | 357 |
|  | 47214 | Retail Trade in Livestock Products | 189 |
|  | 47215 | Retail Trade in Fishery Products | 228 |
|  | 47711 | Clothing Retail Trade | 1690 |
|  | 47712 | Retail Trade in Shoes, Sandals and Other Footwear | 361 |
|  | 47713 | Clothing Complementary Retail Trade | 182 |
|  | 47714 | Retail Trade in Bags, Wallets, Suitcases, Backpacks and others | 225 |
| I | 56101 | Restaurant | 1136 |
|  | 56102 | Food Stores | 1857 |
|  | 56103 | Food Stalls | 355 |
|  | 56104 | Mobile Food Service Activities | 1414 |
|  | 56303 | Café | 143 |
|  | 56304 | Beverage Stalls | 109 |
|  | 56306 | Mobile Beverage Service Activities | 312 |
|  |  | Total | 12850 |

* 1. Methodology
     1. Preprocessing

The pre-processing stage in this research will be carried out with the following steps:

1. Concatenate the main activity description details and the main product description into new details.

Merge example:

Description of main activity: MOBILE BEVERAGE SERVICE ACTIVITIES (PENYEDIAAN MINUMAN KELILING)

Main product description: POPICE

Input: MOBILE BEVERAGE SERVICE ACTIVITIES POPICE (PENYEDIAAN MINUMAN KELILING POPICE)

1. *Lowercasing* is the stage for changing capital letters to non-capital letters.

Input: mobile beverage service activities popice (penyediaan minuman keliling popice)

1. *Stopword removal* is a stage to remove words that appear a lot but do not provide significant meaning, such as "and" and "in”.
2. *Stemming* is the process of returning words to their basic form. Generally, in Indonesian, this process is done by removing affixes. The stopword removal and stemming stages will use the Sastrawi library in Python. In bahasa Indonesia the input will be: *sedia minum keliling popice.*
3. Feature extraction using FastText [12]. Feature extraction is a stage for transforming text data into numerical data. The feature extraction stage is only carried out for models formed using DRF.
   * 1. Modeling
4. DRF

DRF is a machine learning algorithm that learns a function that maps independent variables to response variables based on training data (), Algorithms that do not have assumptions about the form of the function are called non-parametric machine learning algorithms. DRF is a non-parametric machine learning algorithm that can freely learn various functions from training data. The functions in the model are complex and difficult to interpret because they consist of many combinations of single trees. Although difficult to interpret, the DRF classification model can be illustrated in Figure 2.

A diagram of a tree

Description automatically generated

Figure 2 Illustration of the classification model using DRF

As an illustration, the classification model using DRF consists of 500 trees formed in parallel (see Figure 2). Each tree will make predictions based on the probability that data will be predicted as KBLI The final prediction is made by taking the most votes from the 500 trees formed (equation (1)). is the final prediction of the KBLI class based on the independent variable is the result of extracting feature descriptions of the establishment's main activities and main products using FastText is the index of the tree formed . is the KBLI subclasses code, is an indicator function, with a value of 1 if the tree produces a KBLI prediction and 0 if not.

|  |  |
| --- | --- |
|  | (1) |

1. IndoBERT

In the text classification task, BERT uses the final hidden state h of the first token [CLS] to represent the entire text. A simple softmax classifier is added as an output layer of BERT to predict the probability of label c:

Where W is the task-specific parameter matrix, fine-tuning all BERT and W parameters will be done by maximizing the correct label's log probability [13]. Modelling Stages using IndoBERT:

* + 1. Tokenize words using the IndoBERT tokenizer. IndoBERT tokenizer is based on Byte Pair Encoding (BPE) with a total vocabulary of 30,522 [2].
    2. Fine-tuning IndoBERT

The IndoBERT used is IndoBERT large with 335.2 million parameters and 24 layers. Then, an output layer will be added with a softmax activation function, which will calculate the probability of each category of the input data.

* + 1. Model evaluation with accuracy and F1 Score.

A diagram of a block diagram

Description automatically generated

Figure 3 Text classification with IndoBERT

* + 1. Analysis and Evaluation

1. Accuracy

An illustration of the calculation of model evaluation can be referred to in Table 2. If there are k classes, with the predicted value and original value in each class being then accuracy can be calculated using equation (2).

Table 2 Actual and prediction in multi-class classification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Actual | Predicted | | | | |
|  | Class 1 | Class 2 | Class 3 | … | Class k |
| Class 1 | C11 | C12 | C13 |  | C1k |
| Class 2 | C21 | C22 | C23 |  | C2k |
| Class 3 | C31 | C32 | C33 |  | C3k |
| … |  |  |  | … |  |
| Class k | Ck1 | Ck2 | Ck3 | … | Ckk |

|  |  |
| --- | --- |
|  | (2) |

1. F1 Score

The F-1 score combines precision and recall values in one evaluation metric. The F1 score is preferred when an imbalanced class distribution and a balanced measure of precision and recall are needed. Calculating the F1 score in multi-class classification is done by first calculating the F1 score in each class using equation (5), based on recall and precision calculations for each class using equations (3) and (4). The macro average of the F1 score is calculated by averaging the values based on the number of classes using equation (6).

|  |  |
| --- | --- |
|  | (3) |
|  | (4) |
|  | (5) |
|  | (6) |

1. Results

Modelling was carried out using two approaches. The first approach is to create a classification model to predict 16 KBLI subclasses without distinguishing the data section. The second approach is to form a model for each section. The second approach will produce two models: a model for section G, which predicts 9 KBLI subclasses, and a model for section I, which predicts 7 KBLI subclasses. Both approaches aim to explore and compare the evaluation metric of each scenario.

Before the modelling stage, data visualization is carried out to provide an overview of the data. The data for this study is textual, so to visualize it; first, we need the feature extraction step that converts textual data into numerical data. Feature extraction was carried out using FastText Indonesia with dimensions of 300. Next, to visualize data with dimensions of 300, features were reduced to two dimensions using Principal Component Analysis (PCA). The PCA results are then visualized using a dot plot in Figure 4. Although PCA cannot perfectly describe the data's condition, it can still be used to give an overview of the data.

Figure 4 (a), visualization is carried out for all subclasses without separating them by section. Subclasses (indicated by round symbols in different colours) tend to overlap, and it is not easy to separate between each subclass. In Figures 4 (b) and 4 (c), visualization is carried out by separating sections. Figure 4 (b) is a data visualization for section G, and Figure 4 (c) is a visualization for section I. It still can be seen that both section has overlapped units.

|  |  |
| --- | --- |
| (a) | (b) |
| (c) | Figure 4 Visualization Using PCA Based on Feature Extraction Using FastText on (a) 17 KBLI; (b) 9 KBLI in Section G; (c) 7 KBLI in Section I |

The evaluation metrics, which are accuracy and F1 Score for each scenario, are presented in Table 3. These results show that the model with the highest accuracy and F1 Score was obtained by forming two models for each section. As an illustration, by forming two models for each section using IndoBERT, a total accuracy of 0.98 is produced, while if only 1 model is formed, the total accuracy is 0.96. The model produced by fine-tuning IndoBERT has a higher accuracy value and F1 score than the model produced by DRF for overall accuracy and accuracy for each section.

Exploration of the results shows that the model formed for all sections allows prediction errors to occur between sections. Establishment in section I can be predicted as section G, or vice versa; this can be seen in the confusion matrix plot in Figure 5 and Figure 6.

Table 3 Accuracy and F1-Score

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Section | Model | Accuracy | F1Score |
| 1 Model | All | DRF | 0.95 | 0.87 |
|  |  | IndoBERT | **0.96** | **0.92** |
|  | G | DRF | 0.95 | 0.68 |
|  |  | IndoBERT | 0.95 | 0.60 |
|  | I | DRF | 0.93 | 0.72 |
|  |  | IndoBERT | 0.96 | 0.71 |
| 2 Model | All | DRF | 0.96 | 0.91 |
|  |  | IndoBERT | **0.98** | **0.95** |
|  | G | DRF | 0.97 | 0.93 |
|  |  | IndoBERT | 0.98 | 0.96 |
|  | I | DRF | 0.96 | 0.89 |
|  |  | IndoBERT | 0.97 | 0.94 |

Figure 5 compares the actual KBLI with the predicted KBLI from the model. The main diagonal shows correct predictions. The darker the colour, the greater the percentage of correct prediction. For example, Figure 5 (a) shows that DRF predicts 28.57% KBLI 56304 as KBLI 47112. This cross-section prediction exists when using 1 model for all subclasses.

Furthermore, in Figure 5 (b), it can be seen that IndoBERT produces better predictions than DRF. The colours on the main diagonal tend to be darker than DRF; this indicates that the correct prediction for IndoBERT is higher than DRF. The prediction error between sections is also smaller than DRF, marked in light blue in the lower left and upper right corners. For example, the same case, namely, KBLI 56304, predicted as KBLI 47112, did not occur with IndoBERT.

Figure 6 shows the confusion matrix from the second scenario when the models were built for each section. From the confusion matrix, it can be seen that, for section G, KBLI 47713, both models have the lowest percentage of correct prediction. Prediction errors occur by predicting KBLI 47713 as KBLI 47711 and 47112. This error prediction can happen when the establishment has multiple products and the enumerator does not write the specific main product. So, it is essential to write down the main product following the writing rules in the KBLI guidelines.

Apart from that, both models' prediction ability differences can be seen in KBLI 47111. DRF (Figure 6 (a)) shows that 45.85% of KBLI 47111 was predicted as KBLI 47112, while IndoBERT gave higher results, 94.87% of KBLI 47111 was predicted correctly, while the remaining 5.13% predicted as KBLI 47112. These two KBLI are also close together; the only difference is the location of the business.

A graph with numbers and a line of blue squares

Description automatically generated with medium confidence

(a)

A graph with numbers and a bar chart

Description automatically generated with medium confidence

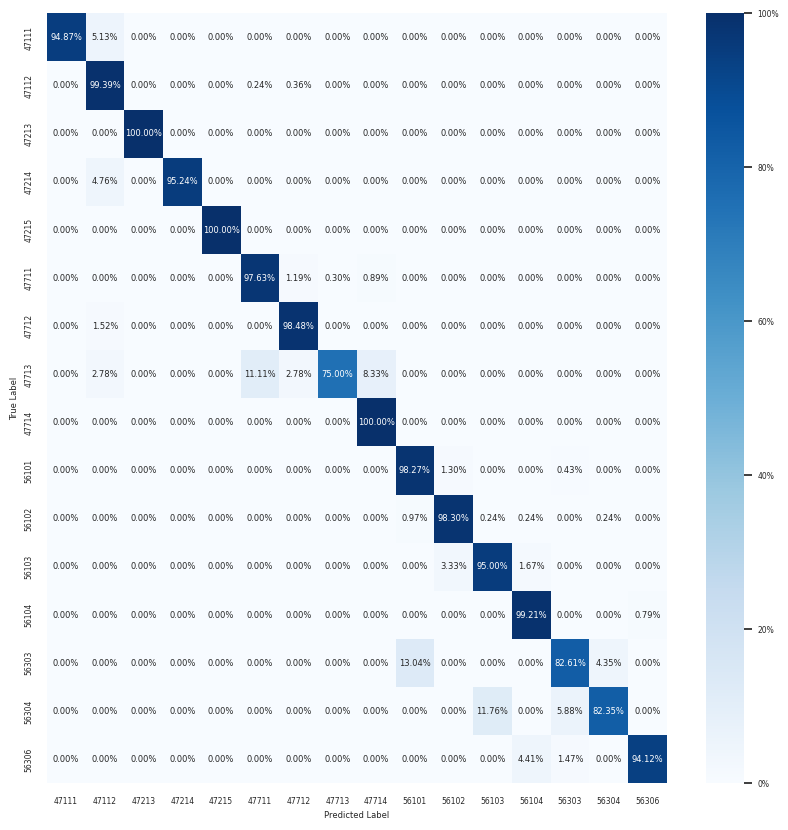
(b)

Figure 5 Confusion matrix from scenario 1: (a) DRF, (b) IndoBERT

A graph with numbers and a number of percentages

Description automatically generated with medium confidence

(a)



(b)

Figure 6 Confusion matrix from scenario 2: (a) DRF, (b) IndoBERT

1. Conclusion

These results show that using machine learning models for industry classification based on textual descriptions of the establishment's main activities and products is feasible. The application of DRF and IndoBERT to classify 5-digit KBLI in sections G and I shows promising results. Two scenarios were formed, which included building a model without differentiating their sections and building a model for each section. For both scenarios, IndoBERT's performance exceeded DRF. IndoBERT has 0.96 accuracy in the first scenario, while DRF is 0.95. In the second scenario, the accuracy of IndoBERT is 0.98 while DRF is 0.96. In this research, forming a model for each section provides better results than forming one model for all sections.

**References**

[1] S. Han, H. Kim, and Y.-S. Lee, “Double random forest,” *Mach. Learn.*, vol. 109, no. 8, pp. 1569–1586, 2020, doi: 10.1007/s10994-020-05889-1.

[2] B. Wilie *et al.*, “IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding,” *arXiv Prepr. arXiv2009.05387*, 2020.

[3] S. Vajjala, B. Majumder, A. Gupta, and H. Surana, *Practical natural language processing: a comprehensive guide to building real-world NLP systems*, 1st ed. O’Reilly Media, Inc., 2020.

[4] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, “Text Classification Algorithms: A Survey,” *Information*, vol. 10, no. 4, 2019, doi: 10.3390/info10040150.

[5] S. Wood *et al.*, “Automated industry classification with deep learning,” in *2017 IEEE International Conference on Big Data (Big Data)*, 2017, pp. 122–129.

[6] S. Slavov, A. Tagarev, N. Tulechki, and S. Boytcheva, “Company Industry Classification with Neural and Attention-Based Learning Models,” in *2019 Big Data, Knowledge and Control Systems Engineering (BdKCSE)*, 2019, pp. 1–7.

[7] M. Dwicahyo and B. Yuniarto, “Deep Learning for Indonesia Standard Industrial Classification,” in *2020 International Conference on Electrical Engineering and Informatics (ICELTICs)*, 2020, pp. 1–6. doi: 10.1109/ICELTICs50595.2020.9315361.

[8] L. Breiman, “Random Forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “{BERT:} Pre-training of Deep Bidirectional Transformers for Language Understanding,” *CoRR*, vol. abs/1810.0, 2018, [Online]. Available: http://arxiv.org/abs/1810.04805

[10] F. Koto, J. H. Lau, and T. Baldwin, “IndoBERTweet: {A} Pretrained Language Model for Indonesian Twitter with Effective Domain-Specific Vocabulary Initialization,” *CoRR*, vol. abs/2109.0, 2021, [Online]. Available: https://arxiv.org/abs/2109.04607

[11] D. R. Faisal and R. Mahendra, “Two-Stage Classifier for COVID-19 Misinformation Detection Using BERT: a Study on Indonesian Tweets.” 2022.

[12] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching Word Vectors with Subword Information,” *Trans. Assoc. Comput. Linguist.*, vol. 5, pp. 135–146, 2017, doi: 10.1162/tacl\_a\_00051.

[13] C. Sun, X. Qiu, Y. Xu, and X. Huang, “How to Fine-Tune BERT for Text Classification?,” in *Chinese Computational Linguistics*, 2019, pp. 194–206.