Semi-Supervised Learning vs. Few-Shot Learning: Which is Better for Sentiment Analysis on Hotel Reviews Towards a Small Labeled Training Data?

Retno Kusumaningrum*, Ahmad Ainun Herlambang, Wafiq Afifah, Adi Wibowo, Sutikno, Priyo Sidik Sasongko
Department of Informatics, Universitas Diponegoro, Semarang, Indonesia

***Abstract*—The massive growth in user reviews on the online travel agent (OTA) website can be automatically processed using sentiment analysis to understand consumer satisfaction or feedback. Sentiment analysis is commonly implemented as a sentiment classification task by applying classical machine learning and deep learning algorithms. However, implementing both strategies has a significant challenge in providing a reliable labeled dataset since labeling is time-consuming and highly resource-intensive. Therefore, this study aims to compare the performance of two learning methods: semi-supervised learning (SSL) and few-shot learning (FSL), since there is still no direct, controlled comparison between both methods. SSL is a learning method that builds a generalized classification model as a refined model based on automatically generated additional labeled data. In contrast, FSL is a learning method that enables a generalized pre-trained model to predict unlabeled data using only a few labeled samples per class. This study evaluates the self-training method on SSL, and the implemented FSL algorithm is Sentence Transformer Fine-Tuning (SETFIT). The results show that implementing FSL (employing only 16 labeled training samples) outperforms SSL with an accuracy improvement of 9.5%. The implementation of SETFIT is very promising as a solution to overcome the limited amount of labeled data in the classification task. Moreover, SETFIT is more adaptable to various low-resource language domains than other, more data-intensive learning approaches.**

Keywords*—Sentiment analysis; hotel reviews; semi-supervised learning; few-shot learning; low-resource language

I. INTRODUCTION

The growth of various IT-based startups in travel, tourism, and hospitality has increased the number of online travel agencies (OTA). A popular OTA feature is hotel bookings. This feature allows users to reserve a hotel and write a review about hotel services or facilities. Subsequently, this condition increases the number of review documents. Hotel reviews have benefits for both hotel users [1], [2] and hotel management [1]. For OTA users, hotel reviews can help them consider which hotel is the most suitable for them based on other users' positive or negative opinions. On the other hand, hotel management can use hotel reviews to manage and improve their current performance to make users more comfortable in the future based on users' negative opinions. The high volume of user review documents can be automatically processed using sentiment analysis to understand the polarity of consumer opinions. Generally, sentiment analysis is performed as a sentiment classification task to predict the sentiment polarities (i.e., positive or negative); thus, building the prediction model needs a set of datasets labeled with their sentiment polarity.

The study about sentiment analysis towards hotel reviews has been widely developed for the Indonesian language. Those studies vary in implementing classical machine learning methods [3], [4] to deep learning methods [3], [5], [6], [7], [8], [9], [10], [11], [12]. Generally, both methods work optimally when a sufficient dataset is available, i.e. 2500 to 5000 review documents. Furthermore, since sentiment analysis is often performed as a classification task, the dataset provided should have been labeled with either positive or negative sentiment polarity. However, manual labeling is time-consuming and highly resource-intensive.

Strategies that can be applied to overcome the problem of limited availability of labeled data are applying other learning methods, namely semi-supervised learning (SSL) and few-shot learning (FSL). As mentioned in study [13], one of the SSL methods that is widely implemented is self-training. A self-training method can leverage the unlabeled training data to refine the classification model performance after it was labeled based on the initial classification model generated by small labeled training data. One of the advantages of the self-training method is the flexibility to be combined with any classification algorithm [14].

On the other hand, several approaches in FSL have been developed by adapting pre-trained language models (PLMs) for specific tasks, such as in-context learning (ICL), parameter-efficient fine-tuning (PEFT), and prompt-based tuning. These approaches typically require specialized infrastructure with limited accessibility and the input of manually generated prompts as part of their training [15]. Therefore, this study employs Sentence Transformer Fine-Tuning (SETFIT), which is competitive with fine-tuning the entire training set despite the three times larger fine-tuned model.

Furthermore, the main contribution of this research is to provide a clearer understanding of which learning strategy is more effective between Semi-Supervised Learning (SSL) and Few-Shot Learning (FSL) in building sentiment analysis models on resource-limited Indonesian-language hotel review data. This study offers a direct, controlled evaluation of the performance of both approaches under minimal labeled-data conditions, thereby providing practical recommendations for developing NLP applications in low-resource scenarios. Thus, this research helps fill the knowledge gap regarding the

*Corresponding author.

selection of the optimal learning method for sentiment classification tasks in the tourism domain.

The structure of this paper is as follows. Section I presents the research background and significance of this study. The comprehensive methodology is clearly explained in Section II. Section III describes the experimental scenarios and their results, including a detailed comparison between scenarios. Section IV is this paper's conclusion and possible future works.

II. METHODOLOGY

A. Dataset Preparation

The dataset used in this research was acquired from the research conducted by study [12]. The raw dataset has two features, i.e., text and class. The text feature includes string letters in uppercase and lowercase, symbols, numbers, and punctuation marks. This text string is in the Indonesian language, but it also has some words or phrases in English, acronyms, and slang based on what the user writes. Those are considered noise and will be cleared during pre-processing. The class feature is a binary class consisting of 0 representing negative sentiment polarity and 1 representing positive sentiment polarity. The dataset comprises 2500 review documents in balanced conditions for both sentiment polarities.

B. Pre-processing

Pre-processing is a phase applied to the dataset before entering the classification phase. Several pre-processing steps will be combined differently for each experiment. Those steps are as follows:

- Case folding is converting all letters to lowercase [16]. This step is conducted because uppercase or lowercase has the same value in sentiment analysis.
- Punctuation mark removal is removing all punctuation marks and symbols [17]. This step is conducted because punctuation marks and symbols have no value in sentiment analysis. In addition, this step also includes URL (Unified Resource Locator) and Tag HTML removal.
- Tokenization is splitting a sentence into individual words or tokens [18]. This step makes further processes, such as feature extraction, more accessible.
- Stop-word removal is the process of removing stop-word, unimportant words [17].
- Stemming is the process of returning words to their origin [19].

There are three different combinations of pre-processing steps for the employed SSL method, as follows:

- PC1 is a pre-processing steps combination including case folding, punctuation mark removal, and tokenization.
- PC2 is a pre-processing steps combination including case folding, punctuation mark removal, tokenization, and stop-words removal.
- PC3 is a pre-processing steps combination including case folding, punctuation mark removal, tokenization, stop-words removal, and stemming.

Implementing FSL involves two pre-processing steps: case folding and punctuation mark removal. It is subsequently as PC4. The implementation of FSL does not involve tokenization, stop-word removal, and stemming steps to avoid the possibility of changing the sentence structure and original meaning of the review data. It is influenced by using the SETFIT framework as the FSL method. SETFIT focuses on sentence processing through the application of sentence transformers so that the structure and meaning of sentences can be maintained when only case folding and data cleaning are applied.

Although the SSL and FSL pipelines differ in several preprocessing steps, this distinction arises from the intrinsic design of the SETFIT framework. SETFIT depends on sentence-level semantic embeddings generated by sentence transformers, which achieve optimal performance when the original sentence structure and contextual cues are maintained. Consequently, only minimal preprocessing, such as case folding and basic punctuation cleaning, is applied to prevent alteration of syntactic patterns or removal of sentiment-bearing elements that influence embedding quality.

On the contrary, SSL employs a token-based classification pipeline that benefits from more extensive preprocessing, including normalization and filtering. Implementing the optimal preprocessing strategy for each paradigm ensures that both methods function under conditions consistent with their respective architectural characteristics. SETFIT's internal analysis shows that applying deeper preprocessing results in only minimal performance shifts, so differences in preprocessing flow do not have substantial implications for the model comparison results.

C. Feature Extraction

Feature extraction uses information from existing data to create vocabulary or corpus so the machine learning algorithm can work. This stage was only involved in SSL implementation, and the extracted feature is TF-IDF. Term Frequency (TF) is the frequent appearance of a term in the document. The more a term appears, the higher its value. Inverse Document Frequency (IDF) is the document's term distribution calculation. The TF-IDF weighting is formulated as shown in Eq. (1):

$$f.idf(t,d) = tf(t,d) \times idf(t) \quad (1)$$

where $tf.idf(t,d)$ represents the weight of each word (term t) in a document d. The term $tf(t,d)$ denotes the weight of *term* t in document d, while $idf(t)$ stands for the inverse document frequency in term t. The calculation of the inverse document frequency is defined in Eq. (2):

$$idf(t) = log\frac{n}{df(t)} + 1 \quad (2)$$

where n is the number of documents, $df(t)$ represents the number of documents consisting term t, and t represents term or word.

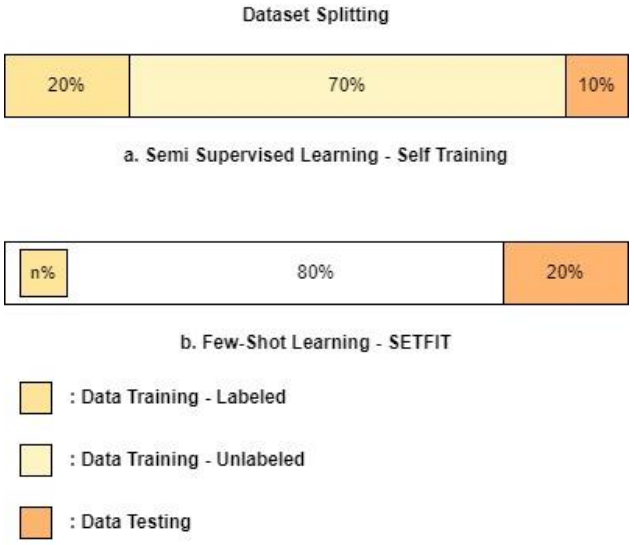

Fig. 1. Dataset splitting in applying (a). SSL and (b). FSL.

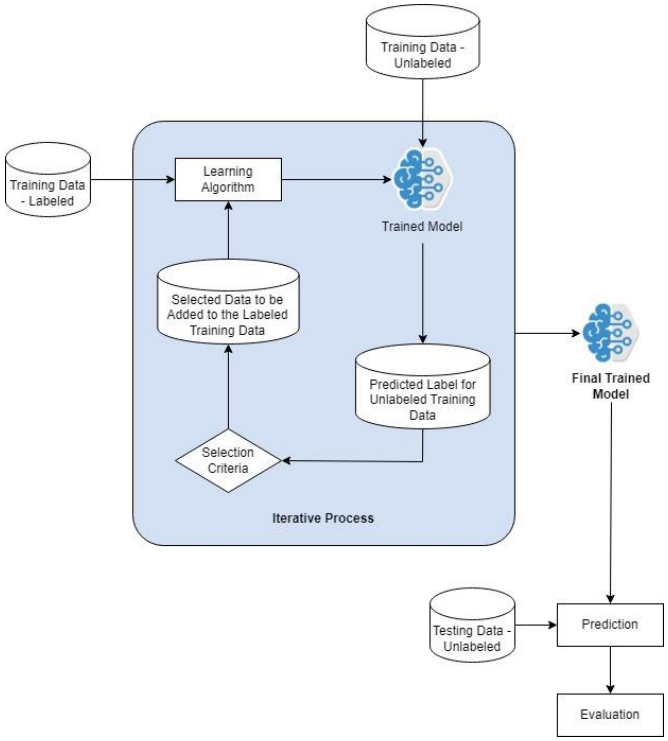

Fig. 2. Architecture of self-training classification model.

D. Building a Sentiment Classification Model

The first step of this stage is dividing the dataset, where there are differences in the implementation of SSL and FSL, as seen in Fig. 1. This study employs K-fold cross-validation for SSL implementation as a validation method to evaluate the performance of the built model by partitioning the dataset into training data and testing data. This research uses 10-fold cross-validation, which means the dataset will be split into ten partitions, and iteratively, each of the nine partitions will be training data while the other one will be testing data. Since this method employs the self-training method in SSL, each training data consists of two parts of labeled training data, and seven other parts are unlabeled. The unlabeled training data will be labeled based on the first generated model and added to the training data, and the training process will be repeated. This process is iterative until a specific condition or predefined number of iterations.

The second data division is involved in applying FSL, where the data is divided into a composition of 80% for the data to be selected as training data and 20% for test data. The training data selection process will be randomly performed on 80% of the previously defined data, and n% of the total data sample per label will be trained (labeled training data). This percentage value will be used as a testing parameter in the research experiment, namely 4%, 6%, and 8%.

Based on the results of respected dataset splitting, the subsequent process is the training process to generate the sentiment classification model. For the implementation of SSL, this study observed the effect of pre-processing steps and the combined classification algorithm on classification performance for the self-training method. As mentioned before, there are three combinations of pre-processing steps, whereas the involved classification algorithms are a support vector classifier (SVC) [20] and stochastic gradient descent (SGD) [21] with a threshold of 0.75. The model building will be conducted with several combinations of parameters based on the three kinds of pre-processing steps mentioned earlier. The architecture of the self-training classification model can be seen in Fig. 2.

Furthermore, this study implements SETFIT as the FSL method. SETFIT is a modification of a transformer model that has been previously trained to minimize the distance between pairs of sentences with similar meanings and maximize the distance between pairs of sentences with different meanings. This model produces sentence representations in dense vectors that the learning algorithms can use by utilizing Siamese network structures and triplet networks to produce semantically meaningful sentence representations [15]. SETFIT consists of two main stages: sentence transformer fine-tuning and classification head training [22]. An overview of the SETFIT architecture can be seen in Fig. 3.

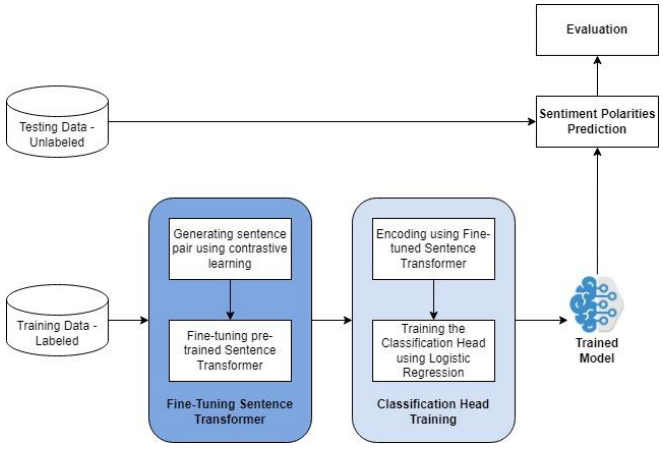

Fig. 3. Architecture of SETFIT model.

As shown in Fig. 3, the sentence transformer fine-tuning stage is divided into two steps: generating sentence pairs using

contrastive learning and performing the fine-tuning process of the pre-trained sentence transformer called paraphrase-multilingualmpnet-base-v2 [23]. Implementing contrastive learning aims to increase the amount of small training data to improve the classifier model's performance. Contrastive learning is learning by comparison, in which the learning process is performed by comparing different input samples, namely comparisons between positive pairs of similar input and negative pairs of dissimilar input [24]. Moreover, the selection of paraphrase-multilingualmpnet-base-v2 is based on this multilingual model from the paraphrase-mpnet-base-v2 version, which has been trained in more than 50 languages, including Indonesian.

There are two steps at the classification head training stage: the encoding process based on the fine-tuned sentence transformer produced in the previous stage and the training of the classification head. The encoding process aims to convert text data into vector representation based on the paraphrase-multilingual-mpnet-base-v2 model that was fine-tuned in the previous stage. The encoding process is performed on validation data from the few-shot learning process to produce output in sentence embedding, which has a richer sentence representation so that each word has a close relationship as a sentence. The next step in the classification head's training process aims to map the previously obtained sentence embeddings into the appropriate classes. This process involves a logistic regression model and is trained to predict data labels correctly. Logistic regression aims to find classification boundary lines that separate data into classes. These boundary lines are represented by regression formulas, where the training process in logistic regression involves using optimization algorithms to find the most appropriate regression coefficients in the regression formula [25]. This research uses the sigmoid function to carry out the binary classification function of 2 sentiment polarity classes.

E. Evaluation

The evaluation compares the result of testing data prediction with the actual class of the data (ground truth). The evaluation metric used in this research is an accuracy value, which can be calculated based on the formula shown in Eq. (3):

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \quad (3)$$

where TP (True Positive) represents the number of correctly predicted data when the model predicts the positive class, and TN (True Negative) denotes the number of correctly predicted data when the model predicts the negative class. Meanwhile, FN (False Negative) refers to the number of incorrectly predicted data when the model predicts the positive class, and FP (False Positive) indicates the number of incorrectly predicted data when the model predicts the negative class.

Accuracy is a representative and stable performance indicator for datasets with a balanced class distribution. Because there is no majority or minority class, the evaluation result indicates the ability to recognize both classes. Furthermore, the study employed the dataset under the same data condition, namely, a balanced dataset. Therefore, accuracy is the right choice when viewed from the perspective of the distribution of data classes in the dataset. Moreover, the two learning strategies or paradigms tested in the study focused on evaluating model behavior using very limited labeled data, rather than evaluating it based on specific classes. Therefore, accuracy is considered a simple, clear, and direct indicator of how each paradigm converts limited data into a correct predictive model.

III. Experiment and Analysis

A. Experimental Setup

This research divides the experiment into three experimental scenarios. The first scenario aims to get the best performance from implementing the self-training classifier as an SSL model by observing the combination of pre-processing and kernel functions. Subsequently, the second scenario aims to obtain the best performance from implementing SetFit as an FSL model by observing the percentage values of training data. In addition, the second experiment performs hyperparameter optimization for each type of training data. This study applies the Optuna framework to search for optimal hyperparameters, using algorithms such as TPE to find the optimal model configuration. The learning rate determines how quickly the model adjusts weights, while the epoch and batch size regulate the intensity and granularity of the optimization process. Parameters such as the number of iterations and the maximum number of iterations control the length of the training process to prevent overfitting and underfitting. The seed setting is necessary to ensure reproducibility of results, while the choice of solver determines the optimization mechanism used during classifier training. Thus, determining the right hyperparameter values is crucial for SetFit to achieve optimal performance under conditions of limited labeled data. The last scenario compares the results from the first and second scenarios.

B. Experimental Results and Analysis

The results of the first scenario can be seen in Fig. 4. A self-training classifier with an SGD tends to produce higher accuracy than an SVC, as depicted in Fig. 4. The accuracy produced by the self-training classifier for both classification algorithms decreased when using four pre-processing stages (PC2) compared to the results when using three pre-processing stages (PC1). The accuracy value subsequently increases when using five pre-processing stages (PC3). However, the highest accuracy values for both classification algorithms were obtained using three pre-processing stages (PC1). The highest accuracy result obtained was 84.52% with a combination of the SGD and 3-stage preprocessing. SGD can produce the best performance because, in theory, even though SGD is simple, SGD can work well in convex optimization [26]. Gradient descent can be used since convexity, defined as the continuity of a convex function's first derivative, ensures that convex optimization problems are smooth and have well-defined derivatives.

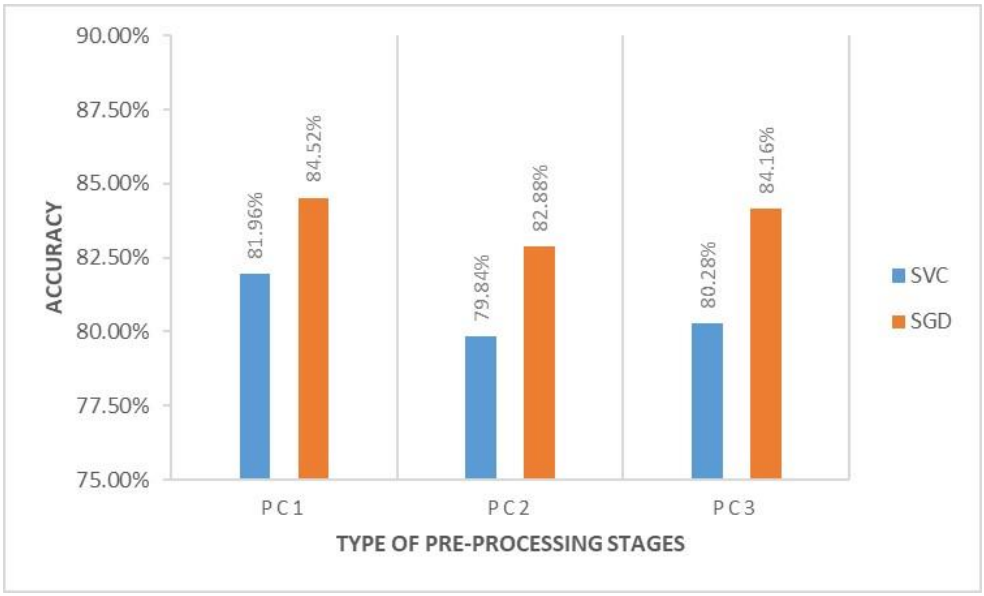

Fig. 4. Result of scenario 1.

TABLE I. RESULT OF SCENARIO 2

Percentage of Training Data for Each Label	Best Hyperparameter							Accuracy
	Learning Rate	*Epoch*	*Batch Size*	*Iteration*	*Seed*	*Max Iteration*	*Solver*	
4%	4.4104e-5	4	4	20	19	235	liblinear	81.25%
6%	4.1996e-5	5	8	10	36	122	liblinear	88.75%
8%	8.094e-5	2	8	10	21	94	liblinear	93.75%

Furthermore, the results of the second experimental scenario can be seen in Table I. The first experiment with eight training samples (4% data samples per label) obtained an accuracy of 81.25%, which is quite good considering the small amount of training data. Subsequently, in the second experiment, with 12 training samples (6% data samples per label), the accuracy value was 88.75%, higher than in the first experiment. The third experiment, with 16 training data samples (8% data samples per label), produced the highest accuracy of the three experiments, namely 93.75%. This result shows that the amount of training data is directly proportional to the accuracy value. The greater the amount of training data, the greater the accuracy value since the model can represent data more efficiently when the number of training data samples increases. In the context of few-shot learning, increasing training samples is still carried out in small data samples, and the model can obtain good accuracy values.

Based on the results from scenarios 1 and 2, it can be concluded that implementing FSL using SETFIT has better performance than implementing SSL using a self-training classifier, with a difference in accuracy value of 9.5%. If compared in terms of the amount of labeled data that must be prepared, SETFIT is more efficient than a self-training classifier (SETFIT), which only requires 12 training samples and can produce an accuracy performance of 88.75%. This accuracy value is higher than the self-training classifier's obtained performance, which requires 500 training samples and only achieved an accuracy of 84.25%. SETFIT's success in outperforming the classical self-training classifier is influenced by the contrastive learning process, which can increase training data by utilizing a pre-trained sentence transformer due to the excellent ability of the pre-trained model to assess similarities between sentences. When using tiny labeled training samples, SETFIT has not been able to produce accuracy values above 90. This weakness may arise because the choice of method for classification head training applied in this research is logistic regression, which has straightforward complexity.

In addition, based on the results from scenarios 1 and 2, it can be said that using FSL (SETFIT) is very efficient for achieving high performance when the amount of labeled data is very small (e.g., <20 samples per class) and computational resources are limited. Meanwhile, the SSL approach is suitable for studies when researchers have access to a sufficient amount of unlabeled data—not a very large amount, but enough to generate pseudo-labels that can enrich the training set. In such conditions, SSL may offer greater benefits than FSL, as the model can gradually expand its training data via a self-training mechanism.

The results of this study not only demonstrate performance differences but also confirm that semantic embedding-based few-shot methods have more efficient learning capabilities than pseudo-labeling in SSL when labeled data is scarce. These findings provide practical guidance for NLP developers in low-resource environments, suggesting that FSLs such as SETFIT are a more reliable and resource-efficient starting point for sentiment analysis in similar domains.

In addition to these advantages, SETFIT also offers greater computational efficiency because it does not require full fine-tuning as conventional transformer models do, making it more suitable for devices with limited resources. Furthermore, SETFIT's characteristic of not relying on large amounts of labeled data makes it more adaptable and easily transferable to other domains with similar low-resource conditions than other learning approaches that are more data- and computationally intensive.

IV. CONCLUSION AND FUTURE WORK

In this study, the performance of SSL and FSL methods for sentiment analysis of hotel reviews with small labeled training data is compared with the help of the gold-standard dataset of Indonesian hotel reviews with two sentiment polarities (positive and negative). Results show that the FSL using SETFIT can outperform the SSL using a self-training classifier algorithm with an accuracy improvement of 9.5%. SSL with self-training obtained an accuracy of 84.25% and employed 500 labeled training samples. In contrast, SETFIT produced an accuracy above 90%, to be precise 93.75%, using only 16 labeled training samples. These findings not only confirm the superiority of SETFIT's results but also demonstrate that semantic representation-oriented few-shot approaches are more effective at learning sentiment patterns than SSL when the number of labels is limited. This provides practical insight that FSLs like SETFIT can be a more efficient and robust starting point for sentiment analysis in low-resource contexts.

Possible directions for future research include exploring other classification methods in classification head training in SETFIT architecture. Thus, the smaller labeled training samples can achieve accuracy above 90%. In addition, this study has a weakness in its dataset scope: it comes from a single OTA platform, limiting linguistic variability and potentially hindering generalization across platforms.

ACKNOWLEDGMENT

This research was supported by the Directorate of Research, Technology and Community Services, Ministry of Education, Culture, Research and Technology, Indonesia, under the “Penelitian Fundamental Regular” scheme, Grant No. PFR Grant 601-54/UN7.D2/PP/VI/2024, dated June 12, 2024. The authors would like to express their gratitude for the financial support that enabled the completion of this study.

REFERENCES

[1] X. Liu, M. Schuckert, and R. Law, “Can Response Management Benefit Hotels? Evidence from Hong Kong Hotels,” J. Travel Tour. Mark., vol. 32, no. 8, pp. 1069–1080, Jul. 2015, doi: 10.1080/10548408.2014.944253.

[2] N. Vo, V. V. Hung, Z. Tučková, N. T. Pham, and L. Nguyen, “Guest Online Review: An Extraordinary Focus on Hotel Users’ Satisfaction, Engagement, and Loyalty,” J. Qual. Assur. Hosp. Tour., vol. 23, no. 4, pp. 913–944, Jun. 2022, doi: 10.1080/1528008X.2021.1920550.

[3] R. Kusumaningrum, I. Z. Nisa, R. P. Nawangsari, and A. Wibowo, “Sentiment analysis of Indonesian hotel reviews: from classical machine learning to deep learning,” Int. J. Adv. Intell. Informatics, vol. 7, no. 3, pp. 292–303, Nov. 2021, doi: 10.26555/ijain.v7i3.737.

[4] T. G. Prahasiwi and R. Kusumaningrum, “Implementation of negation handling techniques using modified syntactic rule in Indonesian sentiment analysis,” in Journal of Physics: Conference Series, Institute of Physics Publishing, Jun. 2019. doi: 10.1088/1742-6596/1217/1/012115.

[5] A. Azhar and M. L. Khodra, “Fine-tuning Pretrained Multilingual BERT Model for Indonesian Aspect-based Sentiment Analysis,” in 2020 7th International Conference on Advance Informatics: Concepts, Theory and Applications (ICAICTA), Mar. 2021. doi: 10.48550/arXiv.2103.03732.

[6] L. M. Cendani, R. Kusumaningrum, and S. N. Endah, “Aspect-Based Sentiment Analysis of Indonesian-Language Hotel Reviews Using Long Short-Term Memory with an Attention Mechanism,” in Emerging Trends in Intelligent Systems & Network Security, M. Ben Ahmed, B. A. Abdelhakim, B. K. Ane, and D. Rosiyadi, Eds., Cham: Springer International Publishing, 2023, pp. 106–122.

[7] J. Fernando, M. L. Khodra, and A. Septiandri, “Aspect and Opinion Terms Extraction Using Double Embeddings and Attention Mechanism for Indonesian Hotel Reviews,” in 2019 International Conference on Advanced Informatics: Concepts, Theory, and Applications, ICAICTA 2019, Aug. 2019. doi: https://doi.org/10.1109/ICAICTA.2019.8904124.

[8] R. A. Genadi and M. L. Khodra, “Opinion Triplet Extraction for Aspect-Based Sentiment Analysis Using Co-Extraction Approach,” J. Inf. Commun. Technol., vol. 21, no. 2, pp. 255–277, Apr. 2022, doi: 10.32890/jict2022.21.2.5.

[9] H. Imaduddin, Widyawan, and S. Fauziati, “Word Embedding Comparison for Indonesian Language Sentiment Analysis,” in 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), 2019, pp. 426–430. doi: 10.1109/ICAIIT.2019.8834536.

[10] R. Jayanto, R. Kusumaningrum, and A. Wibowo, “Aspect-based sentiment analysis for hotel reviews using an improved model of long short-term memory,” Int. J. Adv. Intell. Informatics, vol. 8, no. 3, pp. 391–403, Nov. 2022, doi: 10.26555/ijain.v8i3.691.

[11] P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, “Sentiment Analysis Using Word2vec And Long Short-Term Memory (LSTM) For Indonesian Hotel Reviews,” Procedia Comput. Sci., vol. 179, pp. 728–735, 2021, doi: https://doi.org/10.1016/j.procs.2021.01.061.

[12] R. P. Nawangsari, R. Kusumaningrum, and A. Wibowo, “Word2Vec for Indonesian Sentiment Analysis towards Hotel Reviews: An Evaluation Study,” Procedia Comput. Sci., vol. 157, pp. 360–366, 2019, doi: https://doi.org/10.1016/j.procs.2019.08.178.

[13] F. Hemmatian and M. K. Sohrabi, “A survey on classification techniques for opinion mining and sentiment analysis,” Artif. Intell. Rev., vol. 52, no. 3, pp. 1495–1545, 2019, doi: 10.1007/s10462-017-9599-6.

[14] A. Ligthart, C. Catal, and B. Tekinerdogan, “Analyzing the effectiveness of semi-supervised learning approaches for opinion spam classification,” Appl. Soft Comput., vol. 101, p. 107023, 2021, doi: https://doi.org/10.1016/j.asoc.2020.107023.

[15] L. Tunstall et al., “Efficient Few-Shot Learning Without Prompts,” in The 2nd Workshop on Efficient Natural Language and Speech Processing, Sep. 2022, p. 1. doi: https://doi.org/https://doi.org/10.48550/arXiv.2209.11055.

[16] M. K. Delimayanti, R. Sari, M. Laya, M. R. Faisal, Pahrul, and R. F. Naryanto, “The Effect of Pre-Processing on the Classification of Twitter’s Flood Disaster Messages Using Support Vector Machine Algorithm,” in 2020 3rd International Conference on Applied Engineering (ICAE), 2020, pp. 1–6. doi: 10.1109/ICAE50557.2020.9350387.

[17] W. Etaiwi and G. Naymat, “The Impact of applying Different Preprocessing Steps on Review Spam Detection,” Procedia Comput. Sci., vol. 113, pp. 273–279, 2017, doi: https://doi.org/10.1016/j.procs.2017.08.368.

[18] J. T. Yun, B. R. L. Duff, P. T. Vargas, H. Sundaram, and I. Himelboim, “Computationally Analyzing Social Media Text for Topics: A Primer for Advertising Researchers,” J. Interact. Advert., vol. 20, no. 1, pp. 47–59, Jan. 2020, doi: 10.1080/15252019.2019.1700851.

[19] J. Asian, “Effective Techniques for Indonesian Text Retrieval,” RMIT University, 2007.

[20] C. Zhang, X. Shao, and D. Li, “Knowledge-based Support Vector Classification Based on C-SVC,” Procedia Comput. Sci., vol. 17, pp. 1083–1090, Dec. 2013, doi: 10.1016/j.procs.2013.05.137.

[21] L. Bottou, "Stochastic Gradient Descent Tricks," in Neural Networks: Tricks of the Trade: Second Edition, G. Montavon, G. B. Orr, and K.-R. Müller, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 421–436. doi: 10.1007/978-3-642-35289-8_25.

[22] A. F. Thielmann, C. Weisser, and B. Säfken, "Human in the Loop: How to Effectively Create Coherent Topics by Manually Labeling Only a Few Documents per Class," in Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue, Eds., Torino, Italia: ELRA and ICCL, May 2024, pp. 8395–8405. [Online]. Available: https://aclanthology.org/2024.lrec-main.736/

[23] N. Reimers and I. Gurevych, "Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 4512–4525. doi: 10.18653/v1/2020.emnlp-main.365.

[24] P. Le-Khac, G. Healy, and A. Smeaton, "Contrastive Representation Learning: A Framework and Review," IEEE Access, vol. 8, pp. 193907–193934, Jan. 2020, doi: 10.1109/ACCESS.2020.3031549.

[25] X. Zou, Y. Hu, Z. Tian, and K. Shen, "Logistic Regression Model Optimization and Case Analysis," in The 7th International Conference on Computer Science and Network Technology (ICCSNT), Oct. 2019, pp. 135–139. doi: 10.1109/ICCSNT47585.2019.8962457.

[26] J. Du, "The Frontier of SGD and Its Variants in Machine Learning," in Journal of Physics: Conference Series, Institute of Physics Publishing, May 2019. doi: 10.1088/1742-6596/1229/1/012046.