2024 International Conference on Information Science and Communications Technologies (ICISCT) | 979-8-3315-2792-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICISCT64202.2024.10956585

A BiLSTM-based Sentiment Analysis Scheme for Khmer NLP in Time-series Data

Sokleng Prom[1], Panharith Sun[2], Neil Ian Cadungog-Uy[3], Sa Math[4*], Tharoeun Thap[5*]
[1,2,3]Dept. of Computer Science, Paragon International University, Phnom Penh, Cambodia
[4,5]Ministry of Post and Telecommunications of Cambodia,
Telecommunication Regulator of Cambodia, Phnom Penh, Cambodia
E-mail: [1]promsokleng17@gmail.com, [2]panharith.sun@gmail.com, [3]nuy@paragoniu.edu.kh,
[4]sa-math@mptc.gov.kh, [5]tharoeun-thap@mptc.gov.kh

***Abstract*—This research studies the extraction of sentiments from the Khmer language through machine learning and deep learning methods, specifically the Bidirectional Long Short-Term Memory (BiLSTM) network. To achieve this, it employs a quantitative approach to analyze the result from the different training classifiers, including Support Vector Machine (SVM), Naïve Bayes (NB), Random Forest (RF), K-Nearest Neighbors (K-NN), and the BiLSTM neural network. The study also utilized a pre-trained BERT model on the Khmer language as the embedding model combined with applying preprocessing techniques, such as data cleaning and word segmentation before the classifications. The performance of the models is evaluated using accuracy, precision, recall, and F1-score. The findings reveal that BiLSTM with the contextual embeddings from BERT, achieves the greatest performance resulting in an accuracy rate of 86%, outperforming the traditional machine learning algorithms in classifying Khmer text sentiments into negative, neutral, and positive classes.**

Keywords: Natural Language Processing, Khmer Text, Sentiment Analysis, Word Embedding, BERT, BiLSTM

I. Introduction

Artificial intelligence's increased popularity, especially generative AI like GPT-4, has demonstrated its capability to benefit our lives in many ways. Among various developed models, sentiment analysis remains the up-to-date model for extracting meaning from text and classifying it into different sentiments, primarily the positive and negative polarities. This technology has proven valuable across numerous services in the current technology transformation era. The manual identification of sentiment in textual data is feasible for the limited quantity of texts. However, it becomes challenging when the data increases to millions of records. In the applications of sentiment analysis, large amounts of data could be analyzed accurately and efficiently. In consumer-centric industries, customer feedback analysis and reviews help businesses gain insights to understand customer behaviors, enhancing market research, competitor analysis, and product improvements. In the healthcare industry, sentiment analysis is used to study patient moods and their reactions toward certain drugs. Moreover, sentiment analysis could also serve as an important tool in various contexts including social media monitoring, political and social analysis, stock market trends, reputation management, and other domains. There are a large amount of studies on sentiment analysis methods for commonly spoken languages, including English, Chinese, and Arabic. However, the Khmer language still faces a scarcity of research within the same domain. This could be credited to Khmer being a low-resource language where limited studies are done in the field of NLP [1]. Although translation may be a probable alternative, it is an ineffective technique due to inaccurate translated data and loss of cultural and linguistic nuances [2]. Therefore, a sentiment analysis model specifically trained for Khmer text would open the door for many opportunities, especially in Khmer NLP developments.

The study investigates the development of a Khmer sentiment analysis by classifying it into positive, neutral, or negative sentiments. This model will not cover the processing of textual data in other languages besides Khmer including the mixture of Khmer text with foreign languages, and Romanized Khmer text. Additionally, the study does not cover unrelated issues, including incorrect spelling, grammatical mistakes, illogical sentence structure, and indirect statements. It is assumed that the provided inputs would be grammatically and structurally correct and provide sufficient meaning to be processed through the model and output the sentiments accordingly.

II. Literature Review

Despite the limited number of studies for sentiment analysis on Khmer text, there are regardless some studies that provide insightful information for this research. One project used xlm-roberta-base, a multilingual model based on Facebook's RoBERTa, trained on the "bookmebus" dataset. By conducting a transfer learning method to produce the classification of 5 sentiments, the model was able to produce an accuracy rate of 50% after 10 epochs. Another study investigated co-regulating Transformer models for sentiment analysis. The study pre-trained a BERT model with 100,000 Khmer news articles which resulted in the

Copyright © IEEE–2024 ISBN: 979-8-3315-2792-1

accuracy of 81% through the fine-tuning method. FastText was also applied as an embedding model which yielded lower accuracy at only 77% where both models were used with a deep neural network architecture [3].

With the linguistic similarities between the Khmer and Thai languages, including the derivation from ancient Indian scripts, not using spaces word delimiters, and similar subject-verb-object word order, studies done on Thai sentiment analysis would also provide valuable insights for this research. A study on Thai sentiment analysis for social media employed different machine learning methods including Logistic Regression, Stochastic Gradient Descent, Gradient Boosting algorithm, RF, and Support Vector Classifier which produced the highest result of a 72% accuracy rate [4].

Another study utilized deep learning techniques by training Thai Twitter data on Deep Convolutional Neural Networks (DCNNs) which returned a great accuracy of 75% [5]. Last, another Thai worked on the hotel domain using machine learning algorithms and achieved 89% accuracy through Unigram and Logistic regression [6]. Additionally, research on English sentiment analysis offers further insights for this task. The study conducted a comparison between NB and K-NN classifiers, the NB yields over 80% accuracy rate, outperforming the K-NN approach [7]. Another study focused on using TF-IDF as the vectorization approach combined with a BiLSTM neural network, this method addressed an improvement in accuracy of 92%; however, it was noted that the BiLSTM approach was time-consuming due to its complex architecture [8].

III. The Proposed Method

This paper utilizes the dataset scraped from the Foodpanda and public Bookmebus reviews dataset to research the sentiment analysis task. Table I below illustrates the source and the total row of data collected from each dataset.

Table I: Amount of Datasets

Datasets	Amount
Positive	3938
Negative	6104
Neutral	438
Total	10480

A. *Data Preprocessing*

To study how an individual sentiment is embedded into Khmer text through the training models, it is crucial for the dataset that is used to train the model to go through a preprocessing process to produce a meaningful dataset for the proposed model.

1) *Data Normalization*

Data normalization is also referred to as text normalization in this study and is the process of transforming the raw unstandardized data into a standardized format [9]. The data retrieved are purely in a raw format meaning that irrelevant information including punctuations, emoticons, special characters, and others are present. These components would pose a problem for our tokenization process and increase the unnecessary processing time in the later stages. Therefore, each text row undergoes normalization to remove these elements. For instance, special characters such as "!()[]@#$%^><_" and emoticons are removed, as they do not significantly contribute to sentiment analysis. As some of the data may contain English words that do not have a high impact on this task, those English words are also removed to ensure that the dataset comprises Khmer text only. Lastly, common Khmer punctuations including ។, ។ល។, ៗ, ៕, ៚, ៙, and ៖ are also removed from the dataset.

2) *Stop-Words Removal*

The stop-word corpus containing 385 Khmer stop-words is used for its removal from the dataset in this study. The words in this corpus are the combination of direct translations of English stop-words and manually added Khmer words. This stage allows the data to retain only significant information and reduce the loads of the dataset.

3) *Word Tokenization*

Word tokenization, a process of separating a text corpus into separated words, is a crucial element in most natural language processing tasks including sentiment analysis task. English word tokenization is a simpler process due to the structure of the language being separated by space allowing easier separation of words. However, Khmer text tokenization is a challenging process as it does not have clear delimiters like in English as the words are written from left to right consecutively with optional space and morphemes could be combined to form compound words [10]. To address this complexity of segmenting Khmer text, the researchers will apply 'khmer-nltk', a python package designed specifically for Khmer NLP task, to conduct segmentation of the Khmer word in our preprocessing pipeline for this study.

B. *Word Embeddings*

Word embedding is a fundamental technique used in natural language processing that converts words into machine-readable numeric representations. This study employs a Large Language Model (LLM) which was introduced by Google Research called Bidirectional Representations from Transformers (BERT). The BERT feature-based approach is the embedding model for Khmer text. BERT incorporates bidirectional processing which encodes a text sequence in both forward and backward directions [11]. BERT feature-based approach creates vector representations by placing semantically related words close to each other within the vector space allows training models to understand the semantic relationship between words and enables more efficient information extraction and a greater understanding of sentiment patterns [12].

Copyright © IEEE–2024 ISBN: 979-8-3315-2792-1

This study utilized the pre-trained BERT model from [4] in which the language model was fitted with a large corpus of Khmer newspaper articles scraped from the website with 167896 vocabularies. This technique allows BERT to learn additional patterns, contexts, and numerous Khmer linguistic features making it suitable for extracting the incorporated context of Khmer language in this sentiment analysis dataset. Figure 1 demonstrates the embedding process using the pre-trained BERT model contains five steps.

1) *Special Tokens Addition*

BERT requires inputs to be padded with special tokens: [CLS] at the beginning and [SEP] at the end of sequences. This ensures uniformity across the dataset, with a max sequence length of 64 tokens determined to be the optimal value.

2) *Tokens to IDs Conversion*

The padded tokens are then mapped with their corresponding unique token identifiers (IDs) based on the pre-defined vocabulary list in the LLM. The conversion from word tokens to token IDs is also done to maintain consistency and improve the efficiency of the embedding process.

3) *Sequence Padding*

Sequences shorter than the max length of 64 are padded with zeros until the maximum length is reached.

4) *Attention Masks*

Attention masks are a sequence of 1s and 0s to allow the BERT model to identify the positionings of the input tokens and the padded tokens, respectively. In addition, the attention mask is applied to prevent the model from attending the paddings and only focusing on the actual tokens.

5) *Embeddings Extraction*

Both the input IDs sequence and the attention mask are passed into the pre-trained BERT model where the contextual embedding of the sequences is returned with the embedding size of 768 features.

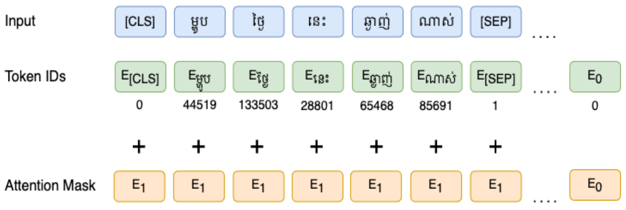

Fig. 1: BERT Model Inputs Visualization

C. *Model*

Sentiment algorithms are among the fundamental elements of this study as they are the components used to perform the sentiment analysis task. Training a model usually involves the learning process of an algorithm to analyze and recognize specific patterns within the provided data and make predictions on unseen data based on the studied patterns. However, with the diversity of training algorithm variations, different training models would lead to different models producing different learning outcomes based on the provided task and input data. To construct the sentiment analysis models, this study will investigate five distinct model training algorithms with four machine learning algorithms and one neural network to analyze the one that offers the highest performance and is most suitable for the proposed study. The machine learning classifiers used in this study were namely the Support Vector Classifier, Multinomial Naïve Bayes, RF Classifier, and K-NN Classifier. The parameters for each classifier are as follows:

- Support Vector Classifier with C=1, decision_function_shape=”ovr”, gamma=0.01 and kernel=”rbf”
- Multinomial Naïve Bayes with its default parameters
- Random Forest Classifier with max_depth=20, min_samples_leaf=10, n_estimators=200
- K-Nearest Neighbor with metric=”manhattan”, n_neighbors=19, weights=”uniform”

Regarding the recurrent neural network configuration, the BiLSTM network architecture is set up as shown in Figure II consisting of two BiLSTM recurrent layers and two dense or fully connected layers. In the fully connected layers, a dropout layer was introduced to discard some information from the previous layers to reduce the chance of overfitting during training. The last dense layer is configured with the “softmax” activation function and outputs a list of confidence scores for each sentiment being negative, neutral, and positive, respectively. As for the hyperparameters setting, the settings shown in Table II were the optimal hyperparameters producing the most satisfactory result post-tuning. As shown in algorithm 1, the training process for BiLSTM contains all the steps mentioned above from preprocessing up to the embedded vectors which are fed into the neural network in a continuous training and validation cycle with the settings presented in Table II.

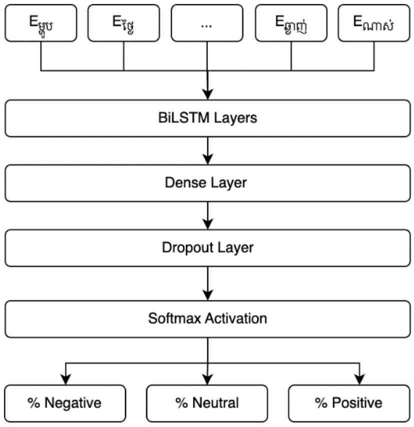

Fig. 2: BERT-BiLSTM Network Architecture

Copyright © IEEE–2024 ISBN: 979-8-3315-2792-1

```
Algorithm 1: BiLSTM Training Algorithm
  Data: Data $X$, labels $Y$, number of epochs $N$, learning rate $\alpha$
  Result: Trained BiLSTM model
  Data Preparation Process;
  $X$ = preprocess($X$);
  $X$ = tokenize($X$);
  $X_t, X_v, Y_t, Y_v$ = splitData($X$, $Y$)
  Embedding Process;
  $tokenizer$ = PretrainedBERTTokenizer();
  $model$ = PretrainedBERTModel();
  $input\_ids_t, attention\_masks_t$ = tokenizer($X_{train}$);
  $input\_ids_v, attention\_masks_v$ = tokenizer($X_{val}$);
  $embedded\_X_t$ = model($input\_ids_t, attention\_masks_t$);
  $embedded\_X_v$ = model($input\_ids_v, attention\_masks_v$);
  BiLSTM Training Process;
  for $epoch$ = 1 to $N$ do
      for each training sample $x_i, y_i$ in $embedded\_X_t, Y_t$ do
          Forward pass through the BiLSTM to compute the output $\hat{y}_i$;
          Compute the loss $L(y_i, \hat{y}_i)$;
          Compute correct training set predictions;
          Backward pass to compute the gradients;
          Update Parameters;
      end
      Compute average training accuracy;
      Compute average training loss;
      Validation Phase;
      for each validation sample $x_i, y_i$ in $embedded\_X_{val}, Y_{val}$ do
          Forward pass through the BiLSTM to compute the output $\hat{y}_i$;
          Compute the loss $L(y_i, \hat{y}_i)$;
          Compute correct validation set predictions;
      end
      Compute average validation accuracy;
      Compute average validation loss;
  end
```

Table II: The Hyperparameter Setup

Hyperparameters	Values
Hidden Size	512
Batch Size	128
Dropout Rate	0.55
Learning Rate	0.00001
Optimizer	Adam
Loss Function	Categorical Cross entropy
Number of Epochs	40

D. Evaluation Metrics

The performance of the trained models is then assessed through the evaluation metrics including accuracy, precision, recall, and f1-score.

1) Accuracy

Accuracy measures the model's predictive capability by calculating the proportion between all correct predictions to the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

TP (True Positive) denotes the number of correctly predicted positive classes, while *TN (True Negative)* expresses the correctly predicted negative class by the model. Conversely, *FP (False Positive)* corresponds to the count of classes incorrectly predicted as positive, and *FN (False Negative)* represents the number of incorrectly predicted negative classes by the model.

2) Precision

Precision is the ratio of the true positive classes to all the predicted positive class. It that shows how a model correctly predicts the target class.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

3) Recall

Recall is the proportion of the true positive classes among the actual positive samples in the dataset. It emphasizes the model's ability to capture all of the actual positive classes.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

4) F1-Score

F1-Score is an evaluation formula that assesses the model's predictive performance by calculating precision and recall.

$$F1\text{-}Score = \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

IV. Experimentation Result

This study evaluated the performance of BERT embeddings on the sentiment analysis dataset using four machine learning algorithms NB, K-NN, RF, and SVM alongside a BiLSTM network on the Kaggle platform with a GPU P100 accelerator. The data was split into training (70%), testing (15%), and validation (15%) sets. The NB had the lowest accuracy (73%) due to its assumption of feature independence, which is violated by the dependencies in Khmer word tokenization and BERT embeddings. The K-NN is slightly better than NB but struggled with high-dimensional BERT embeddings because of the curse of dimensionality, which affected its performance. RF achieved an 81% accuracy rate, higher than both NB and K-NN, but was still impacted by the dimension size of BERT embeddings and the random selection of features. The SVM yielded the highest accuracy (85%) among the machine learning classifiers, benefiting from its ability to handle high- dimensional data and utilize the RBF kernel for complex decision boundaries.

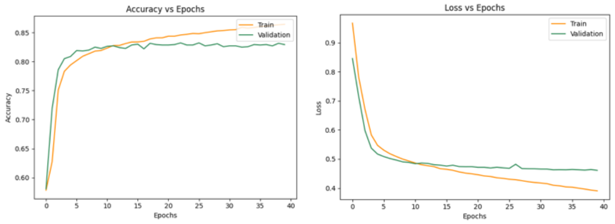

Fig. 3: BiLSTM Training and Validation Accuracy and Loss

The BiLSTM neural network achieved the best result with an 86% accuracy rate by capturing contextual information from BERT embeddings in both directions. Figure 3 shows the accuracy rate and loss during the process of training with the BiLSTM network. Despite its complex architecture, the small dataset size, lack of diversity, and class imbalance limited its performance. The study concludes that BiLSTM's superior performance highlights the importance of capturing long-term dependencies.

Copyright © IEEE–2024 ISBN: 979-8-3315-2792-1

Table III: NB, K-NN, RF, SVM, BiLSTM Performance Results

Approach	Accuracy	Precision	Recall	F1-Score
NB	73%	73%	73%	73%
K-NN	79%	75%	79%	77%
RF	81%	78%	81%	78%
SVM	85%	81%	85%	83%
BiLSTM	86%	84%	86%	84%

In addition, an analysis of the correctness of class predictions was also conducted. The study compared the predicted classes from the finished models against the actual classes, calculating precision, recall, and F1-score as presented in Table III. K-NN, RF, and NB produced comparable results, excelling in predicting positive and negative classes but struggling with the neutral class. NB algorithms had the highest score for neutral class predictions but at the expense of lower scores for other classes. The RF failed to capture neutral sentiment entirely. In contrast, both SVM and BiLSTM achieved similar high scores for negative and positive class predictions, with only slight differences of about 1%.

However, SVM completely failed to identify the neutral class, like RF, whereas BiLSTM demonstrated some capability for detecting neutral sentiments. The failure of RF and SVM to detect the neutral class, along with low results for all models for this class, is attributed to the lack of neutral data in the dataset used in this study.

Table IV: NB, K-NN, RF, SVM, BiLSTM Classification Reports

Approach	Class	Precision	Recall	F1-Score
K-NN	Negative	76%	92%	83%
	Neutral	11%	5%	7%
	Positive	83%	62%	71%
NB	Negative	79%	80%	79%
	Neutral	20%	18%	19%
	Positive	71%	71%	71%
RF	Negative	78%	96%	86%
	Neutral	0%	0%	0%
	Positive	88%	69%	77%
SVM	Negative	84%	96%	90%
	Neutral	0%	0%	0%
	Positive	88%	81%	84%
BiLSTM	Negative	84%	95%	89%
	Neutral	42%	7%	12%
	Positive	89%	80%	85%

V. Result

This study evaluated the performance of various models on sentiment analysis for the Khmer language using BERT embeddings. The machine learning algorithms, including K-NN, RF, and SVM were among the best performers, achieving accuracies of 79%, 81%, and 85%, respectively, as presented in Table IV. NB performed the worst with an accuracy of 73%. BiLSTM showed notable improvement over the machine learning algorithms, reaching an 86% accuracy rate. SVM and BiLSTM were found to perform better than the other methods for this task. In the comparative analysis of class prediction correctness, K-NN, RF, and NB performed well on negative and positive classes but struggled with the neutral class. SVM and BiLSTM displayed similarly high scores, with accuracies over 80%. However, RF and SVM failed to capture the neutral class entirely, whereas BiLSTM and the other models demonstrated some ability to detect neutral sentiments. This deficient performance for the neutral class is attributed to the lack of neutral data in the dataset.

VI. Conclusion and Future work

This research focuses on sentiment analysis for the Khmer language using natural language processing techniques on customer review data. The researchers developed a comprehensive training framework for the Khmer Sentiment Analysis Model, exploring five models: NB, K-NN, RF, SVM, and BiLSTM. Among these, BiLSTM showed the highest accuracy and reasonable speed with an 86% accuracy rate. For the data preprocessing, the study examined techniques like data normalization and stop-word removal. It was found that data normalization alone produced the highest accuracy, as Khmer stop word removal methods, derived from English corpus, did not significantly enhance performance.

Based on the findings, the researchers propose several recommendations for future exploration including experimentation on other training algorithms, embedding models, improved datasets, and analysis of the various linguistic features of the Khmer language. Overall, this research highlights the effectiveness of BiLSTM in Khmer sentiment analysis and emphasizes the importance of data normalization in preprocessing. The proposed recommendations aim to further improve sentiment analysis models by incorporating advanced algorithms, diverse datasets, and Khmer linguistic features.

Acknowledgment

This study was supported by the Ministry of Post and Telecommunications, Kingdom of Cambodia.

References

[1] R. Buoy, N. Taing, and S. Kor, “Joint Khmer word segmentation and Part-of-Speech Tagging using deep learning,” 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2103.16801

[2] E. Bugliarello, “It’s Easier to Translate out of English than into it: Measuring Neural Translation Difficulty by Cross-Mutual Information,” arXiv.org, May 5, 2020. [Online]. Available: https://arxiv.org/abs/2005.02354.

[3] M. R. I. Rifat and A. A. Imran, “Incorporating transformer models for sentiment analysis and news classification in Khmer,” in Lecture Notes in Computer Science, pp. 106-117, 2021. doi: 10.1007/978-3-030-91434-9_10.

[4] S. Srikamdee, U. Suksawatchon and J. Suksawatchon, "Thai Sentiment Analysis for Social Media Monitoring using Machine Learning Approach," 2022 37th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Phuket, Thailand, pp. 1-4, 2022. doi: 10.1109/ITC-CSCC55581.2022.9894882.

[5] P. Vateekul and T. Koomsubha, "A study of sentiment analysis using deep learning techniques on Thai Twitter data," 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE), Khon Kaen, Thailand, pp. 1-6, 2016. doi: 10.1109/JCSSE.2016.7748849.

[6] N. Khamphakdee and P. Seresangtakul, "Sentiment analysis for Thai language in hotel domain using machine learning algorithms.," Acta Informatica Pragensia, 10(2), pp 155–171, 2021. https://doi.org/10.18267/j.aip.155

[7] L. Dey, S. Chakraborty, A. Biswas, B. Bose, and S. Tiwari, "Sentiment Analysis of Review Datasets Using Naïve Bayes‘ and K-NN Classifier," International Journal of Information Engineering and Electronic Business, vol. 8, no. 4, pp. 54–62, Jul. 2016, doi: 10.5815/ijieeb.2016.04.07.

[8] G. Xu, Y. Meng, X. Qiu, Z. Yu and X. Wu, "Sentiment Analysis of Comment Texts Based on BiLSTM," in IEEE Access, vol. 7, pp. 51522-51532, 2019, doi: 10.1109/ACCESS.2019.2909919.

[9] K. Sodimana, P. D. Silva, R. Sproat, T. Wattanavekin, A. Gutkin, and K. Pipatsrisawat, "Text Normalization for Bangla, Khmer, Nepali, Javanese, Sinhala and Sundanese Text-to-Speech Systems," In SLTU, pp 147-151, 2018.

[10] A. Ali, M. Khan, K. Khan, R.U. Khan, and A. Aloraini "Sentiment Analysis of Low-Resource Language Literature Using Data Processing and Deep Learning," *Comput. Mater. Contin.*, vol. 79, no. 1, pp. 713-733. 2024. https://doi.org/10.32604/cmc.2024.048712

[11] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Oct 11, 2018.

[12] D. S. Asudani, N. K. Nagwani, and P. Singh, "Impact of word embedding models on text analytics in deep learning environment: a review," Artificial Intelligence Review, vol. 56, no. 9, pp. 10345-10425, Feb. 2023. DOI: 10.1007/s10462-02