Khmer Sentiment Lexicon Based on PU Learning and Label Propagation Algorithm

CHAO LI and XIN YAN, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Yunnan Key Laboratory of Artificial Intelligence, Kunming, China
GUANGYI XU, Yunnan Nantian Electronic Information Industry Co., Ltd., China
ZHONGYING DENG, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, China
YUANYUAN MO, School of Southeast & South Asia Languages and Culture, Yunnan Minzu University, China

The sentiment lexicon is an important tool for natural language processing tasks. In addition to being able to determine the sentiment polarity of words or phrases, it can assist attribute-level, sentence-level, and text-level sentiment analysis tasks. In light of the fact that tagging data and corpora for the Khmer language are scarce, where most resources related to sentiment lexicons are for English, this paper proposes a method for constructing a sentiment lexicon for Khmer based on **Positive-Unlabeled learning (PU Learning)** and the label propagation algorithm. Sentiment words are first extracted from a corpus using the Spy technique of PU learning method. The main idea is to purify the set of N-class examples, train the MLP classifier, and continuously delete spy words and increase the number of P-class words in the iterative process. Following this, the sentiment polarity of the candidate words is determined. By considering the problem of determining the sentiment polarity of the candidate words as one of calculating its probability distribution, a small number of labeled sentiment words and candidate words are used to construct a graph model. The contextual information of the candidate words is used to construct a simple supplementary graph model of the set of sentiment words through word co-occurrence and triangulation, where this enhances the correlation between data items. The sentiment polarity of the candidate words is then determined through the label propagation algorithm. The results of experiments show that the proposed method can be used to construct a Khmer sentiment lexicon with a small number of labeled data and a small corpus without requiring excessive manual labeling.

CCS Concepts: • **Computing methodologies → Language resources**;

Additional Key Words and Phrases: Khmer sentiment lexicon, PU learning, label propagation algorithm, graph model, Spy technique

This work is supported by National Nature Science Foundation of China via grant 61462055, 61562049.
X. Yan contributed to this study as equally as Chao Li.
Authors' addresses: C. Li, X. Yan (corresponding author), G. Xu, Z. Deng, and Y. Mo, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, Yunnan, 650500, China; emails: lichao.l@foxmail.com, kg_yanxin@sina.com, ntxgy@163.com, 14578626@qq.com, 25080946@qq.com.

2375-4699/2023/03-ART78 $15.00
https://doi.org/10.1145/3564697

ACM Reference Format:
Chao Li, Xin Yan, Guangyi Xu, Zhongying Deng, and Yuanyuan Mo. 2023. Khmer Sentiment Lexicon Based on PU Learning and Label Propagation Algorithm. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 22, 3, Article 78 (March 2023), 18 pages.
https://doi.org/10.1145/3564697

1 INTRODUCTION

Sentiment analysis is an important area of research in the field of natural language processing, and has become more popular with the rise of online social media. A sentiment lexicon contains words that express positive or negative emotions. As an important resource for sentiment analysis tasks, an accurate and comprehensive sentiment lexicon can improve accuracy. It can be used to quickly and accurately determine the sentiment polarity of words, and can assist sentence-level and text-level sentiment analysis tasks. Considerable research has been conducted on the automatic construction of sentiment lexicons. Liu et al. [1] divided the major methods used to construct sentiment lexicons into three categories: knowledge-based methods, corpus-based methods, and a combination of the two. Early research in the area focused on constructing sentiment lexicons by manually determining the sentiment polarity of words or phrases, which usually requires a large amount of manual effort and takes a long time. Moreover, sentiment lexicons constructed using this method are not widely applicable. Methods that use a knowledge base involve constructing the lexicon by using the relationship between synonyms and antonyms in it [2]. The corpus-based approach usually employs a variety of language rules, such as the relationships of inflection and consistency among sentiment words [3].

Economic and cultural exchanges between China and Cambodia have become increasingly frequent in recent years. Khmer, the language of Cambodia, is not very popular in the global context. In the course of its development, influenced by French, Sanskrit and other languages, Khmer is one of the most important languages in East Asia. As Khmer is one of the low-resource languages, the number of learners and users is small, and its word-formation and syntactic features are different from other languages, language barrier has become a major obstacle to cooperation. It is of great significance and practical value to study the Khmer with the method of natural language processing.

At present, the natural language processing technology available for it is relatively poor, and resources for an accurate and rich sentiment lexicon for Khmer are unavailable. The automatic construction of an efficient, accurate, and applicable Khmer sentiment lexicon can benefit research on natural language processing for the language. This can in turn be used for analyses of trends in Cambodia's economic and cultural development, and public opinion monitoring and prediction.

Preliminary work has shown that some results can be used as reference for the automatic construction of a sentiment lexicon. However, such information is used mostly for the automatic construction of English sentiment lexicons, and relevant research on Khmer is scarce. Constructing a Khmer sentiment lexicon through manual labeling requires considerable manual effort and takes a long time. Machine translation is not suitable for Khmer because of its particularity. The selected seed sentiment words have a significant influence on the experimental results when using machine learning and deep learning, and the performance of the model can be improved by using more tagged data. A large corpus is required for training, which is not feasible for Khmer. Inspired by recent research, this paper proposes a method to construct a Khmer sentiment lexicon based on PU learning and the label propagation algorithm. The proposed method can be divided into two steps. (1) Sentiment words are identified from a Khmer corpus through Positive-Unlabeled learning (PU learning) [4]. PU learning is used to train two classification methods for cases of

only positive data and unlabeled data. P-class data are a small number of sentiment words marked manually and U-class data are a group of candidate words obtained from the word segmentation of the Khmer corpus. We identify words that are also sentiment using U-class words. (2) The polarity recognition of these affective words is regarded as equivalent to calculating their polarity distribution, and the correlations between nodes are calculated according to the similarity and intensity of co-occurrence between candidate words and affective words to construct a graph model. The label propagation algorithm [5] is used to estimate the polarity distribution of the candidate words and identify the sentiment polarity of new words.

2 RELATED RESEARCH

A sentiment lexicon is a collection of words expressing emotions. Sentiment words are the most important basic units that carry sentiment information. These words are usually adjectives, adverbs, nouns, or verbs. Researchers have designed many algorithms to automatically construct sentiment lexicons. A common method is to build one based on the available lexical information. Currently used lexicons (such as WordNet and HowNet) mark synonyms and antonyms among words and iteratively extract them according to their occurrence in the lexicon through a small number of seed sentiment words. Kamps et al. [6] used a WordNet-based distance calculation method to determine the sentiment polarity of adjectives, and Hassan et al. [7] used the Markov random walk model on a word correlation graph to predict the sentiment tendencies of words. Esuli et al. [8] used supervised learning methods to identify the sentiment polarity of sentiment words, and Dragut et al. [9] used a set of empirical rules of inference to deduce the sentiment tendencies of other words.

The corpus-based method is a common method of using the connectives of certain linguistic rules to identify the sentiment words and sentiment polarity in a given corpus. Turney et al. [10] proposed the SO method **(Sentiment orientation, SO)**. Firstly, the set of positive and negative seed sentiment words was constructed, and the sentiment lexicon was constructed by setted the threshold and calculated the difference between the candidate words and the positive and negative seed words. Kanayama et al. [11] used such conjunctions as AND, BUT, and OR to identify the sentiment consistency between and within sentences. Words that appeared consecutively were assumed to have the same polarity unless transition words were encountered, in which case the sentiment polarity changed. Krestel et al. [12] used LDA to divide a corpus into different topics, and then used judgment feature analysis and potential topic extraction to automatically generate a topic-related sentiment lexicon.

Recent studies have made use of neural networks and word embedding, and have regarded the construction of a sentiment lexicon as a classification task. Tang et al. [13] expanded a sentiment lexicon by using the softmax classifier through a large number of training word embeddings of text. Hamilton et al. [14] combined domain-specific word embedding with tag dissemination to construct a domain-specific sentiment lexicon. Bravo-Marquez et al. [15] used an emoji corpus and supervised learning methods to classify words, and Vo [16] proposed a simple and effective 2D neural network-based emotion representation in which each word was matched with a 2D vector, and the network was associated with sentiment polarity in each dimension to determine the sentiment polarity of words. Deng et al. [17] proposed a novel sparse self-attention LSTM (SSALSTM) to construct a large-scale sentiment lexicon.

In recent years, more and more attention has been paid to the construction of Non-English language and domain specific sentiment lexicon. Wu et al. [18] constructed a domain-specific sentiment lexicon on Weibo by using the knowledge of vocabulary-emotion knowledge, emotional similarity knowledge, and prior knowledge of existing emotion dictionaries. Due to the lack of complete knowledge base and corpus, there is little research on language rules in low-resource language. The cost of constructing sentiment lexicon by manual method is huge, and the domain

coverage is small. In the existing research on the construction methods of low-resource language sentiment lexicon, the expansion of sentiment lexicon is usually realized by machine translation and cross-language projection, which usually requires efficient and accurate machine translation tools and accurate high-resource language data. Abdaoui et al. [19] expanded the French sentiment lexicon through the semi-automatic translation and expansion of synonyms in the English NRC lexicon. The target language is limited by high-resource language data, so it cannot effectively solve the problem of part-of-speech conversion after translation. Liu et al. [20] expanded the sentiment lexicon by manually selecting the seed sentiment words from HowNet, and then using Word2Vec to train word embeddings, the similarity between seed words and candidate words was calculated to constructed graph model and propagation matrix, and the sentiment polarity of candidate words was obtained.

Existing methods have achieved many gratifying results, some languages have a complete and open semantic knowledge base (such as English and Chinese), a general lexicon can be constructed by mining the relationship between words. In the study of low resource languages, based on English Wordnet, India, Myanmar, Thailand, Laos, Vietnam, and other countries have participated in the research of Asian Wordnet system. But Khmer lacks a complete semantic knowledge base, and the manual construction of semantic knowledge base requires a lot of manpower and material resources, these methods are not ideal in the construction Khmer sentiment lexicon.

Past research can be used as reference when constructing a sentiment lexicon. Currently available methods mostly use the lexical information of resources for the sentiment lexicon, and improve classification performance by increasing the number of tagged data items. Even given the lack of corpora, the high cost of manual tagging, and the absence of an available lexicon to use, the above process can be applied to Khmer by using seed words to expand the sentiment lexicon. Fewer positive and negative seed words lead to a smaller coverage of the expanded sentiment lexicon, such that many sentiment words cannot be found. The amount of tagging data for Khmer is small, manual tagging is expensive, and the effect of training a neural network is not ideal. The corpus contains positive, negative, and non-sentiment words; because of the small number of positive and negative sentiment words in the available lexicon, the extraction of candidate sentiment words is regarded as a positive unmarked learning task, and the semi-supervised method is used for it. To determine the sentiment polarity of the candidate words, owing to a lack of *a priori* knowledge of their meanings, the available tagged Khmer lexicon is relatively small. If the correlation between candidate words is calculated directly, the problem of sparse data arises. The tag propagation algorithm proposed in this paper is used to judge the sentiment polarity of the candidate words. By using the contextual relationships among the candidate words, triangulation is used to expand the marked thesaurus and accurately calculate the sentiment polarity of words. This helps strengthen the relationship between nodes in the graph model, and allows us to more accurately judge the sentiment polarity of the candidate words. Using these two steps, this paper proposes a method to construct a Khmer sentiment lexicon through PU learning and the label propagation algorithm.

3 RESEARCH FOUNDATIONS

This section introduces the method to construct a Khmer sentiment lexicon based on PU learning and the tag propagation algorithm. It consists of two steps: (1) We identify sentiment words from the Khmer corpus by using the espionage PU learning method classified by a **Multi-Layer Perceptron (MLP)**. (2) By calculating the cosine similarity and intensity of co-occurrence between sentiment words and candidate words, the degree of correlation between them is calculated, and the label propagation algorithm is used to estimate the polarity distribution of the candidate words to judge their polarity.

3.1 PU Learning Method for Extracting Candidate Sentiment Words

PU learning is a method to learn a classification from a small number of positive samples and unlabeled samples. Given a dataset P of a specific class and a set of unlabeled data U, which contains P-class and non-P-class (called N-class) data, the data in U are divided into P-class and N-class data through the construction of a classifier by using the P set and the U set. Many researchers have studied PU learning [21–23]. A commonly used method for it is a two-step method: (1) Determine reliable counterexamples and identify a group of N counterexamples that are very different from positive examples in the unlabeled dataset U. (2) Use the positive samples and reliable counterexamples to train classifiers, even if P and RN need to be used to build classifiers (such as the SVM and NB) to classify the remaining unlabeled data U-RN. The methods for extracting reliable counterexamples in the first step usually include espionage technology in the S-EM [4], the Rocchio classifier, and PNHL technology. The second step involves training the classifier flexibly, such as by using the SVM, NB, and the biased SVM.

Liu [23] applied PU learning to a text document classification task and used the **support vector machine (SVM)** as a classifier for document classification using the espionage method. Zhang et al. [24] classified suggestion sentences through PU learning, and Maekawa [25] clustered nonlinear attribute graphs through PU learning. Jiang [26] embedded learning words from a scarce resource language through PU learning, and Kiryo [27] proposed a method of PU learning based on non-negative unbiased estimation. Because the Khmer corpus is small, the cost of manual tagging is high, the training and test sets for it are small, and a large sentiment lexicon for finding and correcting the results of classification is not available, the method for extracting candidate sentiment words based on PU learning used here involves increasing the number of positive examples during iterations, which improves the classifier. Then, when judging the sentiment polarity of the candidate words, we carry out correction and screening to reduce the labor needed to build a Khmer sentiment lexicon as much as possible.

Wang [28] used the PU learning method to construct a sentiment lexicon, and achieved good results by applying the enhanced MLP instead of the SVM for classification. In contrast to the traditional method, the main idea here is to treat the labeled dataset U as containing N-class data with noise (instances of hidden class P), and to train the classifier to delete words in U that may belong to class P. By using the Spy technique, randomly selected $\alpha = 10\%$ from the positive sample P to form the spy set S, which is added to the untagged dataset U to form the set "Us." Then, the MLP classifier is trained using datasets P (assigned category label + 1) and Us (assigned category label − 1). The resulting classifier assigns a probability to each instance in U and S. After each iteration, words with probabilities higher than θ are considered to belong to class P; instances in U and S that have a probability higher than θ are deleted, and the updated sets U and S are used for the next iteration. When the stopping criterion (ψ) is met, the iteration stops.

The Khmer sentiment lexicon dataset P marked by experts is very small, and has an insufficient number of positive examples and too many counterexamples. The sentiment intensity of a small number of words in P is thus weak, and cannot be used to identify instances of the hidden class P in set U. Therefore, data from class P are used for further training. In the iterative process, P-class data with high scores are selected and added to set P, and the training continues through this updated set P and Us. After each iteration, instances in the U set that are most likely to belong to P are deleted according to the set threshold, pick the top η probability value among the deleted instances, assign them a category label of + 1, add them to the P set, and carry out the next iteration with the Us set through the updated P set. Meanwhile, we considered that before the iteration stops, if the P set is expanded without limitation, then the results deteriorate as more and more wrong instances are added as iterations progress. And then, in addition to selecting data with higher score values to add to the P set, at the same time, a stop iteration standard (τ), ψ and τ both stop iteration criteria are

set through the verification set, they are interrelated, and the best combination is found through the verification set. Gaussian fitting is used to set a threshold to control the probability of deleting an instance. As this is conservative, the instance deleted from U is very likely to belong to P. A total $\eta = 15\%$ of instances with the highest scores are regarded as those of P (10% and 20% of the results were similar in the experiment).

(1) The probability that each spy word $w_i \in S$ in set S is assigned to P by the classifier. These probabilities are fitted through Gaussian fitting, and the parameters of the Gaussian distribution are obtained by maximum likelihood estimation, as shown in Formulae (1) and (2):

$$\delta = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

$$\varepsilon^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)^2 \tag{2}$$

The setting of the threshold θ is shown in Formula (3):

$$\theta = \delta + \varepsilon \tag{3}$$

(2) The MLP is a forward-propagation network model composed of hierarchical perceptrons with a multi-layered structure. It is trained by the back-propagation algorithm. The MLP has three layers. The first takes the vector of each word as input and outputs a 50D vector. The second layer takes the output of the previous layer as input to produce a 2D output. The first and second layers both use the ReLU activation function. The 2D output of the second layer is connected to five POS features to form a 7D feature vector as the input to the third layer. The POS features are divided into five categories: (verbs, nouns, adjectives, adverbs, and others). For instance, [1,0,0,0,0] signifies a verb. The third layer uses the Sigmoid activation function.

3.2 Label Propagation Algorithm to Judge the Polarity of New Words

3.2.1 Label Propagation Algorithm. The **label propagation algorithm (LPA)** [5] is a semi-supervised method of learning based on graphs. The algorithm spreads label-related information on a similarity graph between nodes. It constructs the graph by using the relationship between labeled and unlabeled samples. The model uses the label-related information of marked nodes to predict that of unmarked nodes. Compared with methods that use semantic knowledge [6, 9] and neural networks [13] to identify word polarity, the label propagation algorithm needs to only use a small amount of labeled data. Through correlation with unlabeled data, the assignment of labels to unlabeled data has the characteristics of high efficiency, a good classification effect, and strong practicability. The tag propagation algorithm is suitable for the task of judging the polarity of candidate words.

Label propagation algorithms have been widely used in community detection, text classification, and relationship extraction. Ren et al. [29] used tag propagation algorithms to classify sentiments in languages with scarce resources, and Huang et al. [30] used constraint-related information to constrain tag propagation algorithms to automatically construct sentiment dictionaries in specific fields. Zhao et al. [31] used contextual and topic-related features as well as the social relationship between users to calculate sentiment scores, and constructed a sentiment lexicon for Weibo through a tag propagation algorithm. Hong et al. [32] used information from HowNet to judge the sentiment polarity of new words through a tag propagation algorithm.

Few sentiment words in Khmer are marked by experts as benchmark words. Some benchmark words and candidate words are far from each other in the text, and many candidate words fail to find the benchmark words that co-occur with them. This results in data sparseness, and reduces the accuracy of recognition. Because the context of candidate words contains non-benchmark words that may have sentiment inclinations, the contextual information of the candidate words is used in addition to the benchmark words and candidate words. Compared with prevalent methods, the proposed method uses two types of nodes to build a graph model, and two similarity functions to define the weights of edges between the different types of nodes. This strengthens the association between nodes and improves the accuracy of recognition.

A simple Khmer sentiment lexicon is constructed by the triangulation method to expand the benchmark words, and the co-occurrence of benchmark words and candidate words is used to improve the accuracy of recognition. Because the parts of speech and words have different meanings in different contexts, the results of a simple translation are unsatisfactory. The triangulation method is based on the following assumption: Two languages A and B are translated into a target language. If the translation result is the same, the word sense may be similar to the two source languages. This method is better than simple translation, and is convenient for extension and correction. SentiWordNet and HowNet are used to extract words with strong sentiment inclinations, and Google Translate is used to translate them into Khmer. Overlapping words in Khmer are filtered out after the English-to-Khmer and Chinese-to-Khmer translations. However, the corresponding words of the source language have different sentiment polarities. For example, words are strongly positive in English, and when they are positive in Chinese, we assign them to positive categories and delete ones that are not sentiment. When the polarities of the two source languages are different, we correct them by manual inspection. We then compare them with sentiment words in Khmer marked by experts, remove repetitive words, and use the remaining as benchmark words.

By calculating the similarity between co-occurring words of the candidate and the benchmark words, the probability distribution of the polarities of the co-occurring words is calculated, and co-occurring words with clear sentiment tendencies are selected as nodes. The semantic similarity and strength of co-occurrence of the benchmark words, candidate words, and co-occurring words are calculated as the weights of edges between nodes to construct a graph model. Finally, the label propagation algorithm is used to estimate the polarity distribution of the candidate words to determine their sentiment polarity.

3.2.2 Graph Model. Define an undirected graph model $G = (X, W)$, $X = \{x_1, x_2, x_3 \ldots x_i\}$ that represents a collection of nodes in a graph, where W represents a set of weights of edges connecting nodes in the graph, and the weight is the correlation between the nodes. There are two types of nodes in the graph model. One is the candidate word set $X_s = \{s_1, s_2, s_3 \ldots s_i\}$ extracted from the corpus in the first step. The other is the set of co-occurring words $X_t = \{t_1, t_2, t_3 \ldots t_i\}$ of candidate words. Co-occurring words are words that co-occur with the candidate words in a fixed-length sliding window, including benchmark words and non-benchmark words. The benchmark words consist of two parts: Khmer sentiment words marked by experts and sentiment words expanded by triangulation method; Non benchmark words are words without manual labels in co-occurrence words. The correlation between nodes is denoted as $w_{ij}(w_{ij} \in W)$, and there are no edges between the co-occurring words. The graph model built is shown in the Figure 1.

3.2.3 Non-benchmark Word Polarity Distribution. Firstly, the semantic similarity between the non-benchmark word set $T_c = \{c_1, c_2 \ldots c_i\}$ in the co-occurring words and each benchmark word in the benchmark word set $T_d = \{d_1, d_2 \ldots d_i\}$ is calculated by calculating the vector cosine similarity

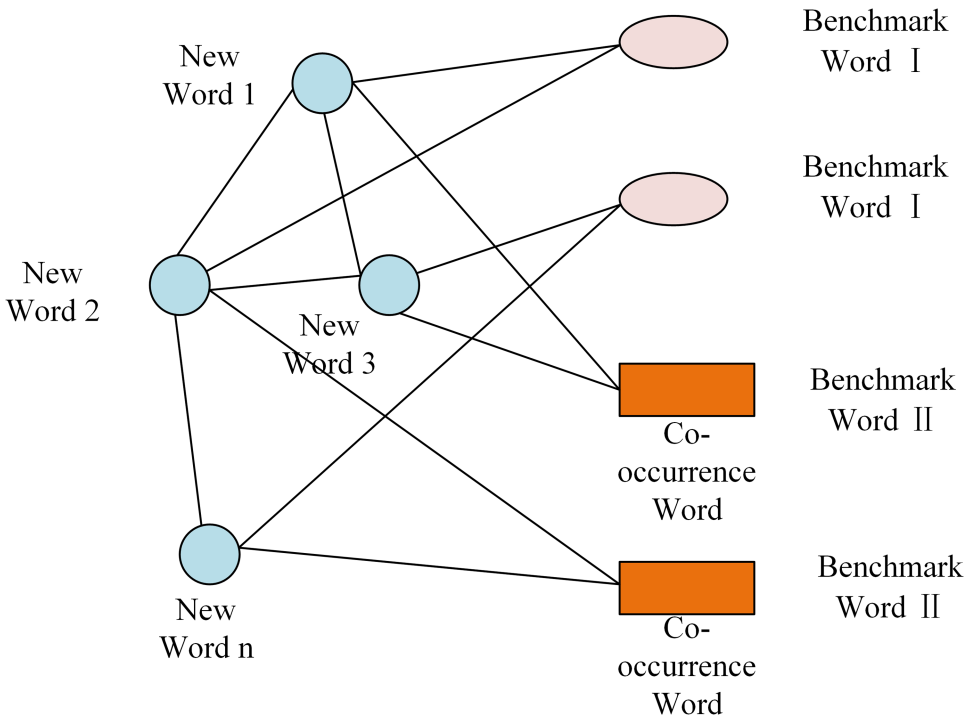

Fig. 1. Structure of the graph model.

method. The calculation method is as follows: Formula (4).

$$SIM(c_i, d_i) = cos(\boldsymbol{c_i}, \boldsymbol{d_i}) = \frac{\boldsymbol{c_i} \cdot \boldsymbol{d_i}}{|\boldsymbol{c_i}| \times |\boldsymbol{d_i}|} \tag{4}$$

where $\boldsymbol{c_i}, \boldsymbol{d_i}$ respectively represent the vectors of the co-occurring word c_i and the benchmark word d_i.

Following this, the semantic similarity is used to determine the distribution $c_i(l)$ of the non-benchmark word c_i in the polarity l of the co-occurring words, where $l = \pm 1$ represent positive and negative sentiment tendencies, and $l = 0$ indicates no sentiment tendency. A threshold λ is obtained and set through experiments. If the similarity between the co-occurring word c_i and each benchmark word d_i is less than λ, the co-occurring word is considered to have no sentiment tendency, $c_i(l = 0) = 1$. If not, calculate according to formula (5).

$$c_i(l) = \begin{cases} \frac{\sum SIM(c_i, d_i^+) > \lambda}{\sum SIM(c_i, d_i) > \lambda}, & l = 1 \\ 0, & l = 0 \\ \frac{\sum SIM(c_i, d_i^-) > \lambda}{\sum SIM(c_i, d_i) > \lambda}, & l = 0 \end{cases} \tag{5}$$

where d_i^+ represents words with positive affective tendencies in the benchmark word set and d_i^- represents those with negative affective tendencies.

In the label propagation algorithm, the label of a known node is transferred to other nodes through the similarity between them. The greater the similarity between nodes, the easier the label propagation. We use the sliding window method to extract co-occurring words, and then calculate the sentiment polarity distribution of the co-occurring words, delete non-benchmark words with

no sentiment tendency, reduce their influence on the correct rate of polar label propagation, and increase the calculation speed.

3.2.4 Correlation Calculation. We use two different similarity functions to calculate the similarity between the benchmark word and the non-benchmark word in the candidate words and the co-occurring words, as the weight of the edges between nodes. The first category is the similarity between the candidate word and the base word in the co-occurring word. The weight of the edge in the graph model is the correlation between the nodes, which is expressed by cosine similarity as shown in Formula (6).

$$w_{ij} = \cos(\boldsymbol{d_i}, \boldsymbol{s_j}) = \frac{\boldsymbol{d_i} \cdot \boldsymbol{s_j}}{|\boldsymbol{d_i}| \times |\boldsymbol{s_j}|} \tag{6}$$

$\boldsymbol{d_i}, \boldsymbol{s_j}$ represent the vector representation of the Khmer sentiment word corrected by the expert and the candidate word based on the PU learning extension, respectively.

The other is the similarity between candidate words and non-benchmark words. Co-occurring words are obtained by defining a window of length K and using the sliding window method to extract words. Words in the sliding window are considered to be co-occurring, and the strength of co-occurrence is inversely proportional to the distance between them. In the graph model, the correlation between the edges can be expressed as:

$$w_{nm} = \sum_{k=1}^{K} F(s_n,\ c_m) \times G(s_n,\ k,\ c_m) \tag{7}$$

$F(s_n, c_m) = 1 + K - k$ is the co-occurrence strength between the candidate words and the co-occurring words, co-occurrence intensity indicates that co-occurrence intensity increases with the distance between two words in the same sliding window. When a sliding window of length K has a distance of k in the related document, the number of co-occurrences of s_n and c_m is expressed as $G(s_n, k, c_m)$.

The weights between the types of nodes are then normalized as follows:

$$w'_{nm} = \frac{w_{nm} - \min(w_{nm})}{\max(w_{nm}) - \min(w_{nm})} \tag{8}$$

In summary, the correlation between nodes in the graph model $w_{ij},\ w'_{nm} \in W$.

3.2.5 Determining Sentiment Polarity of Candidate Words. In the labeling algorithm, nodes update their labels according to the labels of adjacent nodes. The greater the similarity between adjacent nodes, the greater the weight of influence of the given label, and the easier it is to spread the label. During the iterative process of label propagation, labels of the labeled data remain unchanged and those of unlabeled data are continually updated. When the labels of the unlabeled data stop changing, the iterations end. At this time, the probability distributions of similar nodes tend to be similar, and the sentiment polarity distribution of the unlabeled data is obtained to identify their sentiment polarity and conclude label propagation.

We establish an objective function based on the above analysis and optimize it to minimize the value of H:

$$H = \sum_{\substack{x_i \in X_s \\ x_j \in K(s_i)}} w_{ij} \|r_i - r_j\|^2 + \beta \sum_{x_i \in X_s} \|r_i - U\|^2 \tag{9}$$

$$\|r_i - r_j\|^2 = \sum_{l} (r_i(l) - r_j(l))^2 \tag{10}$$

$$\|r_i - U\|^2 = \sum_y (r_i(l) - U)^2 \tag{11}$$

The polarity distribution of node x_i is $r_i(l)$. Node x_j is the K neighboring node of x_i, and its polarity distribution is $r_j(l)$; β is determined by experiment. In the objective function (9), the first term represents the square error of the polarity distribution between the candidate word and its K neighboring nodes. The more similar the polarity distributions of nodes are, the greater the degree of similarity between them. Thus, the value of this term needs to be small. If the candidate word node is not similar to the co-occurring word node, its polarity distribution is considered to be close to the uniform distribution U; that is, the second term represents the square error between the polarity distribution of the candidate word node and the uniform distribution. It can be shown that the objective function is a convex function.

There is a global optimal solution for the objective function, and an analytic solution cannot be obtained. Thus, the optimal solution is obtained through an iterative updating method. We iteratively update the polarity distribution of the candidate word nodes through the following formula:

$$r_i^{(m)} = \begin{cases} c_i(l), & x_i \in T_c (T_c \in X_t) \\ v_i(l), & x_i \in T_d (T_d \in X_t) \\ \frac{\varphi(l)}{K_i}, & x_i \in X_s \end{cases} \tag{12}$$

$$\varphi_i(l) = \sum_{x_j \in k(x_i)} w_{ij} r_j^{m-1}(l) + \beta U \tag{13}$$

$$K_i = \beta + \sum_{v_j \in k(x_i)} w_{ij} \tag{14}$$

In Formula (12), the numerator indicates that the polarity distribution of the K neighboring words of the candidate word is propagated to the candidate words according to similarity, and the denominator is the parameter β plus the sum of the similarity among the K neighboring words of the candidate word, $c_i(l)$ is the polarity distribution of the non-benchmark words in the co-occurring words, and $v_i(l)$ is the polarity distribution of the benchmark words.

Through iterative update, the final label propagation yields the polarity distribution of the candidate word nodes, which is recorded as $o_i(l)$. According to the distribution of the positive and negative polarities of the candidate words, a linear classifier is used to identify them as follows:

$$o_i = \begin{cases} 1, & o_i(l=1) - o_i(l=-1) > \xi \\ 0, & |o_i(l=1) - o_i(l=-1)| < \xi \\ -1, & o_i(l=-1) - o_i(l=1) > \xi \end{cases} \tag{15}$$

where $o_i(l=1)$ denotes $o_i(l)$ as a positive probability, $o_i(l=-1)$ is expressed as a negative probability, and the threshold ξ is obtained by experiments.

4 EXPERIMENT AND DISCUSSION

4.1 Extracting Candidate Sentiment Words

4.1.1 Experimental Data and Steps. We used a corpus of Khmer text crawled from the Cambodia Daily and websites of other publications to expand the sentiment lexicon. A total of 1,400 documents were collected containing 3,564 Khmer sentiment words as marked by experts. The laboratory Khmer word segmentation tool was used to segment the text and obtain the POS tag of each Khmer word [33]. Figure 2 shows the input interface of Khmer sentences ("Government and local NGO representatives had drafted a petition yesterday calling for a halt to the Don Sahong hydropower project which planned to build 1.5 km from Cambodia-Laos border and believed to

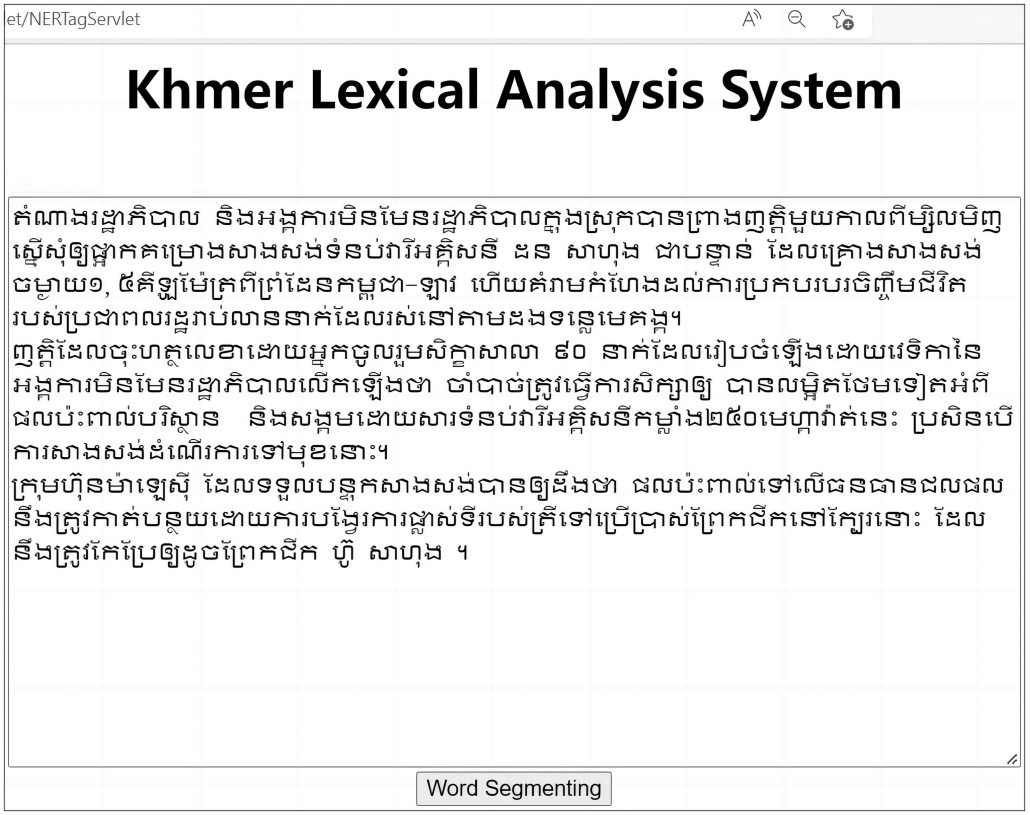

Fig. 2. Khmer sentence input interface.

threaten the Livelihoods of millions of people living along the Mekong River." "Petition that signed by 90 workshop participants organized by NGO Forum pointed out that further environmental and social assessments of the 250 MW hydropower plant is necessary if construction work is to continue.." "The contractor of the hydropower station, the Malaysian company, pointed out that to reduce the impact on fish resources, it is necessary to use the surrounding raceway, just like the Hou sahong Canal."). The word segmentation tool is shown in the Figures 3 and 4.

Word2Vec was used to train the word vectors, and each had 200 dimensions.

We randomly extracted 300 texts for word segmentation and to train the word vectors, and deleted words that appeared in the sentiment lexicon labeled by the experts. A total of 65,492 unlabeled words were used as experimental data. In summary, there were a total of 3,564 words in the set P and 65,492 words in set U. The verification set contained 200 words randomly selected from P and 900 randomly selected from U. The test set contained 2,000 words randomly selected from the remainder of U (65,492 − 900 = 64,592). A total of 428 sentiment words had been manually corrected by experts, of which 216 expressed positive inclinations and 212 expressed negative inclinations. A total of 3,364 words were left in set P, and the 62,592 words in U were used as the training set for the classifier. We randomly extracted α = 15% of the words from P (selected according to past work [23]; the experiment tested α = 10% and 20% as the spy, but the effects were not very different) as the spy set S (504), and used PS and U+S to train the MLP classifier. The criteria for stopping the itera-tions were ψ (=25%) and τ (=30%) as determined by the validation set. $|S| < \psi \cdot |S|$ and $|P| < \tau \cdot |P|$ respectively indicate that the iterations had stopped when the

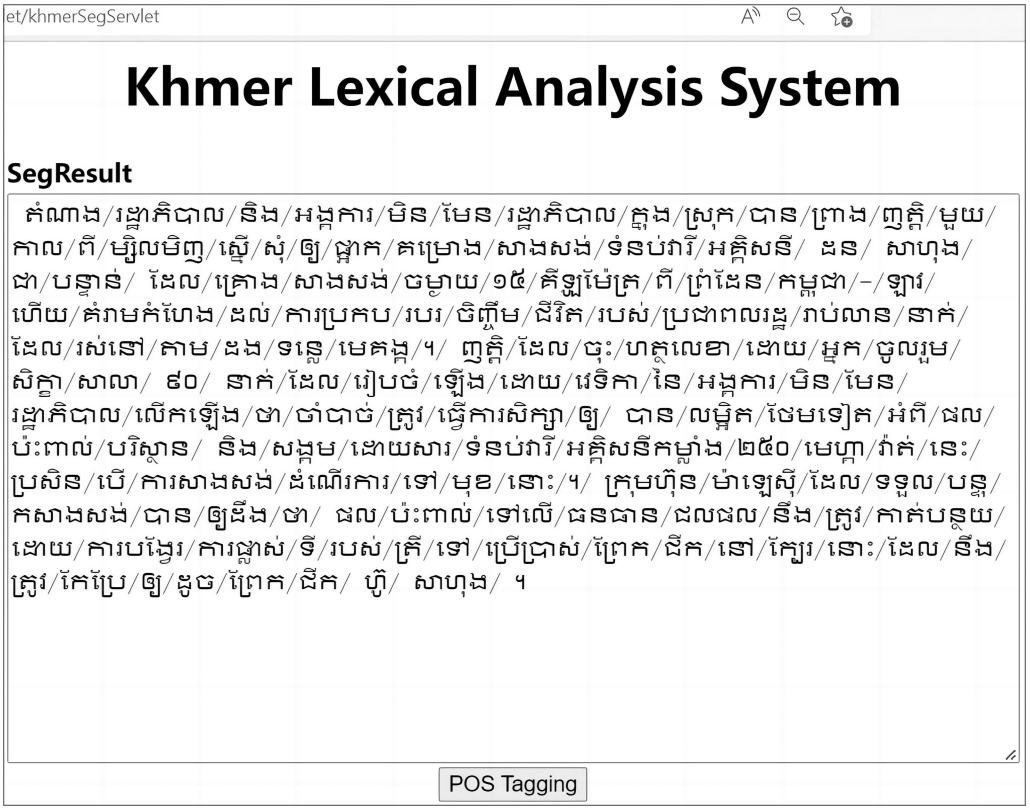

Fig. 3. Results of word segmentation using the Khmer word segmentation tool.

number of words in the spy set and those in the original set P, and the iterations were stopped when either was satisfied. We use *Precision*, *Recall*, and *F1* score for evaluation.

Precision, Recall, and the F1 score were used as evaluation metrics for the experimental results.

4.1.2 Experimental Results and Analysis. We verified the feasibility and effectiveness of the proposed method for extracting candidate sentiment words based on PU learning through comparative experiments. The experiments compared the proposed method with the following three baseline methods on the same dataset:

(1) PMI: The classic PMI method selects experts to mark positive and negative sentiment words in an sentiment word set as benchmark words, and calculates the PMI score between the candidate words and the benchmark words.
(2) S-SVM: Liu et al. [23] used Spy technique to identify N reliable types of counterexamples RN from the set U, trained SVMs to assign probabilities to each instance, and determined the probability threshold. Instances below the probability threshold were divided into those of class N. In this method, the 5D POS feature and a 200D vector were connected to form a 205D feature vector.
(3) SS-MLP: The general idea is to delete instances of type P from sets U and S, regardless of the impact of the large difference in number between positive and negative examples on the results [28].
(4) SP-MLP: This is the PU learning method proposed in this article.

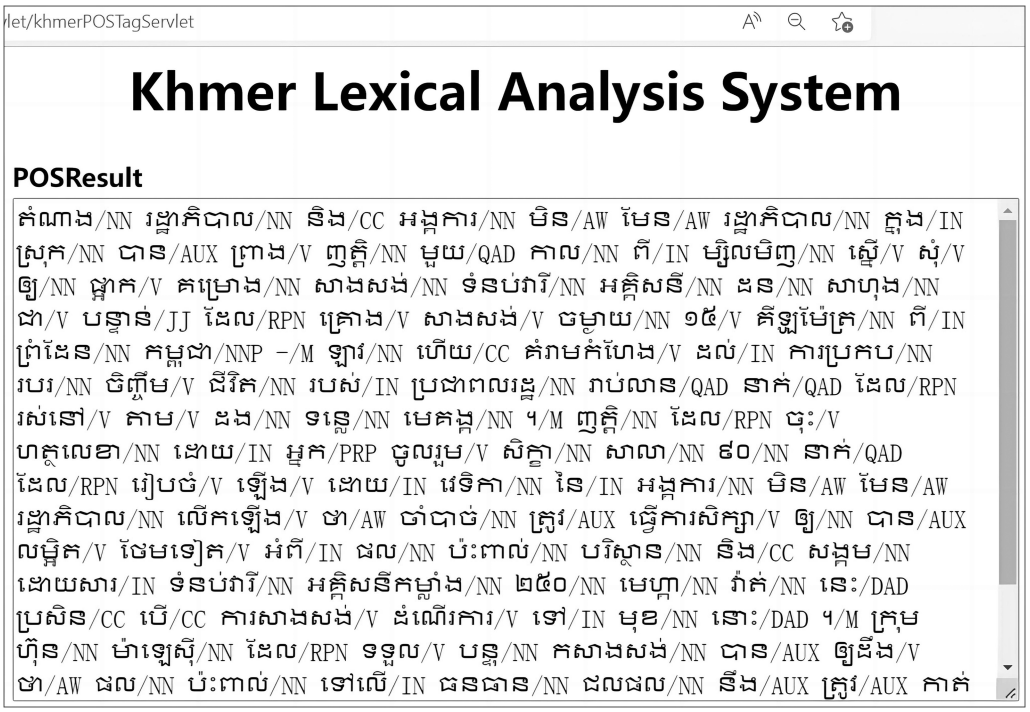

Fig. 4. Results of parts-of-speech tagging using the Khmer word segmentation tool.

Compared with the text method, the S-SVM, S-MLP, and SS-MLP randomly selected $\alpha = 15\%$ of words from the set P and the spy set S. The results are shown in Tables 1 and 2.

Table 1 shows the results for the PMI, S-SVM, S-MLP, and SS-MLP methods. PMI performed better than S-SVM. The best method was not very different from the S-MLP in terms of results. However, the performance of S-MLP declined as the number of iterations increases. The best results of each of the three methods were also worse than those of the SS-MLP. Compared with the S-SVM and S-MLP methods, because the set RN of trusted N-type instances obtained in the first step of PU learning was too small to represent all N-type instances, the iterative operation increased the number of reliable counterexamples, and N thus came to contain hidden P-type instances, which degraded performance of traditional PU Learning. On the contrary, the SS-MLP method deleted instances that might have belonged to P from the unmarked set U to purify N. Before the stopping condition for the iterations was satisfied, the accuracy and recall of the SS-MLP method after each iteration as well as its F1 score were constantly improving. Compared with the S-MLP method, its accuracy increased from 52.9% to 66.1%, recall increased from 51.1% to 57.9%, and the F1 score increased from 51.9% to 61.7%. The SS-MLP outperformed the S-SVM and S-MLP because they added classified N-type instances to RN in each iteration. This had a positive effect on the results initially, but as the number of iterations increased, their performance declined because many hidden P-type instances with insufficient sentiment tendencies were added from U to the RN set. As the iterations progressed, more and more instances of error were added and degraded performance. The SS-MLP method deleted instances of P from U to purify N, and the threshold θ was very conservative. Only instances of P were deleted, and U was regarded as the data for N. The purer it was, the better the classification effect, until the stopping condition was met.

Table 2 shows the results for the proposed SP-MLP method. Compared with the SS-MLP method, which yielded the best performance among the conventional methods, its precision increased to

Table 1. Results of the PMI, Traditional S-SVM, S-MLP, and SS-MLP Methods

	Iteration	P	R	F1
PMI		54.8	50.3	52.5
S-SVM	0	44.8	40.4	42.5
	1	48.4	44.4	46.3
	2	51.4	47.9	49.6
	3	47.6	44.9	46.2
S-MLP	0	51.5	50.0	50.7
	1	53.1	51.5	52.3
	2	55.1	52.0	53.5
	3	52.9	51.1	51.9
SS-MLP	0	54.1	50.5	52.2
	1	55.8	51.5	53.6
	2	58.5	52.5	55.3
	3	60.9	53.7	57.1
	4	62.8	55.2	58.8
	5	64.5	56.7	60.3
	6	**66.1**	**57.9**	**61.7**

Table 2. Results of the Proposed SP-MLP Method

	Iteration	P	R	F1
SP-MLP	0	54.1	50.5	52.2
	1	56.2	51.9	54.0
	2	59.2	53.2	56.2
	3	61.9	54.8	58.1
	4	64.1	56.5	60.1
	5	65.7	58.6	61.9
	6	**67.9**	**60.2**	**63.6**

67.9%, the recall increased to 60.2%, and the F1 score increased to 63.6%. The SS-MLP deleted words that were most likely to be instances of P from U according to a conservatively set threshold. The number of marked instances in P and unmarked instances in U were very different; that is, the number of positive examples was small, and the effect of extracting words with low sentiment strength was not good. The proposed method added the top 15% of the instances most likely to be P type to P, and continued to iterate after the update. The number of words deleted from the spy set increased in each iteration. By increasing the number of P-type instances, it is better to find the hidden P-type instances in the U set. Because the number of elements of P was increased without limit, errors accumulated, and the iterations were stopped when one of the two stopping conditions was met. At this time, the performance of the proposed method was superior to that of the other methods. This is because the PU learning method used to extract the candidate words proposed here yielded a better effect for Khmer with fewer annotated data items and a small corpus.

4.2 Polarity Classification

4.2.1 Experimental Data and Steps. We used the candidate word set extracted by the first experiment sp-mlp (a total of 10,954 candidate words), their context, the sentiment word set marked by experts and the sentiment word set obtained by triangulation method as experimental data. The

sentiment word set marked by experts contained 690 positive sentiment words and 890 negative sentiment words, and there were 480 positive words and 570 negative words that obtained by the triangulation method after removing repeated sentiment words. The test set are sentiment words marked by experts in the first experiment, including 428 sentiment words, 216 positive words, and 212 negative words. Through experiments, we selected the best $K = 6$ to construct the K neighbor graph. Along with the method proposed in this paper and the best method proposed in the literature [29], a label propagation algorithm that considered only the benchmark words was used in a comparative experiment. The methods were used to determine the sentiment polarity of the candidate words. Accuracy of positive and negative sentiment words were used as evaluation metrics of the experimental results.

4.2.2 Experimental Results and Analysis. The influence of the threshold values λ and ξ on the experimental results was analyzed through experiments and their optimal values were determined. λ was the threshold when calculating the sentiment polarity distribution of co-occurring words, the value of which will affected the distribution of the candidate words. According to the probability distribution at different polarities, the polarity of the candidate words was affected by ξ. The influence of the threshold valueswere interrelated. Through experiments, the best experimental effects were obtained at $\lambda = 0.2$ and $\xi = 0.5$, and these were used for subsequent experiments.

The experiments compared the proposed method with the following baseline methods on the same dataset:

(1) PMI: Randomly select the marked Khmer benchmark word as the seed word, directly calculate the similarity between the seed word and the candidate word, and then calculate the difference between the PMI of the candidate word and the positive seed word and the negative seed word to determine the candidate words sentiment polarity.
(2) SVM: Use the labeled Khmer benchmark words to train the SVM classifier, and then perform sentiment classification on the candidate words.
(3) LP: The label propagation algorithm described in paper [29].
(4) **SSWEu (Sentiment-Specific Word Embedding Unified model):** The model takes into account both the context syntactic information and affective information of the sentence learning word embedding for classification. [13].
(5) **SS&W2V-LP (Sentiment Seed words and Word2Vec–Label Propagation):** Select the sentiment seed words manually, Word2Vec is used to train word embeddings on corpus, and then the label propagation algorithm is used to obtain the sentiment polarity of the candidate words. [20].
(6) **B-LP (Benchmark words-Label Propagation):** Ablation Experiment, only considers the label propagation algorithm of benchmark words.
(7) **O-LP (One weight-Label Propagation):** Ablation Experiment, the weight of edge is only defined by the co-occurrence strength.
(8) **C-LP (Context-Label Propagation):** Proposed methods.

The experimental results in Table 3 show that compared with the traditional PMI and SVM methods, the accuracy of the label propagation algorithm is improved. Compared with the method that uses the co-occurrence relationship between candidate words and benchmark words, the LP method has a lower accuracy rate. The reason is that the LP method does not consider the relationship between words in the sentence. In addition to selecting a more suitable calculation method for similarity between words, the Context-Label Propagation algorithm proposed in this paper uses the co-occurrence relationship between candidate words and benchmark words. In addition, the relationship between candidate words and context is used, and the label word set is

Table 3. The Performance of Methods

	Accuracy of positive sentiment words	Accuracy of negative sentiment words
PMI	58.8	54.7
SVM	60.9	56.3
LP	62.4	57.6
SSWEu	63.8	57.9
SS&W2V-LP	64.2	58.2
B-LP	64.3	58.5
O-LP	65.1	58.9
C-LP	**67.7**	**60.1**

expanded through triangulation to supplement the graph model, and the correlation between nodes is enhanced. The experimental results show that the accuracy rates reached 67.7% and 60.1%. The proposed label propagation algorithm effectively solves the problem of sparse data in sentiment polarity classification by using the contextual information. The classification recognition is improved, and it can be effectively applied to the sentiment polarity recognition of Khmer candidate words with less annotated data and scarce corpus.

The proposed SP-MLP extracted canditate sentiment words from unlabeled data. A total of 10,954 words were extracted after cleaning and filtering out duplicates. Then we classified the sentiment polarity of the candidate words.There are 1,386 positive sentiment words and 1,672 negative sentiment words obtained by expert tagging, triangulation method and test data. The proposed label propagation algorithm identified 1,878 positive sentiment words and 2,442 negative sentiment words through a linear classifier. A total of 7,378 words from the Khmer sentiment lexicon were obtained including 3,264 positive sentiment words, 4,114 negative sentiment words.

5 CONCLUSION

This paper proposed a method to construct a Khmer sentiment lexicon based on PU learning and the tag propagation algorithm. We first extracted sentiment words from a corpus, selected the sentiment word set marked by experts as the positive example set, randomly selected the spy word set from it, and train an MLP classifier. The main idea is to purify the set of N-type examples, continually delete elements of the spy word set and increase the size of the set of P-type words in an iterative process. Following this, we determined the sentiment polarity of the candidate words, constructed a graph model between them and the hand-annoted sentiment word set, used the contextual information of the associated candidate words to find the co-occurring word set of the candidate words, and constructed the sentiment word set through the triangulation method. The set was used to assist in judging the sentiment polarity distribution of the co-occurring word set, and part of it was also used as co-occurring words. There was an edge between a word with a clear sentiment tendency and the candidate word. By calculating the probability distribution of the sentiment polarity of the candidate words, the label propagation algorithm was used to judge their sentiment polarity. The experimental results showed that the proposed method can be used for the construction of a Khmer language sentiment lexicon. It also solves the problems of a small amount of tagged data for Khmer, a small corpus, and sparse data in the recognition of candidate word emotion models, and improves accuracy. The construction of a more accurate Khmer sentiment lexicon through a semi-supervised method is useful for subsequent research on natural language processing for Khmer.

However, the proposed method also has some shortcomings. When extracting candidate words through the PU learning method, it does not consider whether the instances highly likely to belong to P deleted from U contain noise. On the other hand, in the judgment of sentiment polarity of candidate words, we just consider the contextual information of candidate words, the information of the candidate words and the syntactic structure in the context are not considered. In future research, we will seek to address these limitations.

REFERENCES

[1] B. Liu. 2012. *Sentiment Analysis and Opinion Mining*. San Rafael, CA, Morgan & Claypool Publishers, (2012).

[2] M. Q. Hu and B. Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, ACM, (2004), 168–177.

[3] V. Hatzivassiloglou and K. R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*. Stroudsburg. PA, USA, Association for Computational Linguistics, 174–181.

[4] Bing Liu, Wee Sun Lee, Philip S. Yu, and Xiaoli Li. 2002. Partially supervised classification of text documents. In *Proceedings of the Nineteenth International Conference on Mach ine Learning (ICML'02)*.

[5] X. Zhu and Z. Ghahramani. 2002. Learning from Labels and Unlabeled Data with Label Propagation [J]. Tech. Rep., Technical Report CMU-CALD-02.107, 2002.

[6] J. Kamps, M. Marx, R. J. Mokken, and M. de Rijke. 2004. Using WordNet to measure semantic orientation of adjectives. In *Proceedings of LREC*.

[7] A. Hassan, A. Abu-Jbara, R. Jha, et al. 2011. Identifying the semantic orientation of foreign words[C]. *Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers*. Association for Computational Linguistics, (2011).

[8] A. Esuli and F. Sebastiani. 2007. PageRanking WordNet Synsets: An application to opinion mining. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*. Prague, Association for Computational Linguistics, 424–431.

[9] E. C. Dragut, C. Yu, P. Sistla, et al. 2010. Construction of a sentimental word dictionary[C]. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM'10)*. Toronto, Ontario, Canada, (October 26–30, 2010), ACM.

[10] P. D. Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews[J]. arXiv preprint cs/0212032, (2002).

[11] H. Kanayama and T. Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA, Association for Computational Linguistics, 355–363.

[12] R. Krestel and S. Siersdorfer. 2013. Generating contextualized sentiment lexica based on latent topics and user ratings. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*. New York, NY, ACM 129–138.

[13] D. Y. Tang, F. R. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin. 2014. Learning sentiment-specific word embedding for Twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Baltimore, Maryland, USA, Association for Computational Linguistics, 1555–1565.

[14] W. L. Hamilton, K. Clark, J. Leskovec, et al. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora[C]. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*. NIH Public Access (2016), 595.

[15] F. Bravo-Marquez, E. Frank, and B. Pfahringer. 2015. Positive, negative, or neutral: Learning an expanded opinion lexicon from emoticon-annotated tweets[C]. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15)*. AAAI Press.

[16] D. T. Vo and Y. Zhang. 2016. Don't count, predict! An automatic approach to learning sentiment lexicons for short text[C]. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* 2 (2016), 219–224.

[17] D. Deng, L. Jing, J. Yu, et al. 2019. Sparse self-attention LSTM for sentiment lexicon construction[J]. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27, 11 (2019), 1777–1790.

[18] F. Wu, Y. Huang, Y. Song, et al. 2016. Towards building a high-quality microblog-specific Chinese sentiment lexicon[J]. *Decision Support Systems* 87 (2016), 39–49.

[19] A. Abdaoui, J. Azé, S. Bringay, et al. 2017. Feel: A French expanded emotion lexicon[J]. *Language Resources and Evaluation* 51, 3 (2017), 833–855.

[20] H. T. Liu, J. C. Zhu, X. Y. Liu, et al. 2019. Expansion of sentiment lexicon based on label propagation[C]. *2019 15th International Conference on Semantics, Knowledge and Grids (SKG)*. IEEE, 145–152.
[21] X. Li and B. Liu. 2003. Learning to classify texts using positive and unlabeled data. In *Proceedings of the Eighteenth International Joint Conference on Artifical Intelligence* 3 (2003), 587–592
[22] Cho-Jui Hsieh, Nagarajan Natarajan, and Inderjit S. Dhillon. 2015. Pu learning for matrix completion. In *ICML*. 2445–2453.
[23] Bing Liu. 2011. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Second Edition. Springer.
[24] Zhang Pu, Liu Chang, and Li Xiao. 2019. Suggestion sentence classification method based on PU learning. *Journal of Computer Applications* 39, 3 (2019), 639–643.
[25] S. Maekawa, K. Takeuch, and M. Onizuka. 2018. Non-linear Attributed Graph Clustering by Symmetric NMF with PU Learning[J]. (2018).
[26] Chao Jiang, Hsiang-Fu Yu, Cho-Jui Hsieh, and Kai-Wei Chang. 2018. Learning word embeddings for low-resource languages by PU learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1(Long Papers)*. Association for Computational Linguistics, 1024–1034.
[27] R. Kiryo, G. Niu, M. C. D. Plessis, et al. 2017. *Positive-Unlabeled Learning with Non-Negative Risk Estimator*[J]. (2017).
[28] Y. Wang, Y. Zhang, and B. Liu. 2017. Sentiment lexicon expansion based on neural PU learning, double dictionary lookup, and polarity association[C]. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.
[29] Ren Yong, N. Kaji, N. Yoshinaga, et al. 2011. Sentiment classification in resource-scarce languages by using label propagation[C]. In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation (PACLIC'25)*. Singapore, (Dec. 16–18, 2011), 420–429.
[30] S. Huang, Z. Niu, and C. Shi. 2014. Automatic construction of domain-specific sentiment lexicon based on constrained label propagation[J]. *Knowledge Based Systems* 56 (2014), 191–200.
[31] C. Zhao, S. Wang, and D. Li. 2019. Exploiting social and local contexts propagation for inducing Chinese microblog-specific sentiment lexicons[J]. *Computer Speech & Language* (2019).
[32] Hong Xudong, Yu Zhengtao, Yan Xin, et al. 2015. Emotional polarity recognition of new words based on label propagation algorithm[J]. *Journal of Frontiers of Computer Science & Technology* 9, 12 (2015), 1506–1512.
[33] Pan Huashan, Yan Xin, Zhou Feng, et al. 2016. A Khmer word segmentation and part-of-speech tagging method based on cascaded conditional random fields[J]. *Journal of Chinese Information Processing* 30, 4 (2016), 110–116.

Received 13 January 2021; revised 16 August 2022; accepted 14 September 2022