Journal of Network Intelligence ©2018 ISSN 2414-8105(Online)
Taiwan Ubiquitous Information Volume **3**, Number **2**, May **2018**

Using CCLM to Promote the Accuracy of Intelligent Sentiment Analysis Classifier for Chinese Social Media Service

Yi-Jen Su
Department of Computer Science and Information Engineering
Shu-Te University
No.59, Hengshan Rd., Yanchao, Kaohsiung County 824, Taiwan
iansu@stu.edu.tw

Hsin-Te Wu
Department of Computer Science and Information Engineering
National Penghu University of Science and Technology
No.300, Liuhe Rd., Magong City, Penghu County 880, Taiwan
wuhsinte@gms.npu.edu.tw

Yue-Qun Chen
Department of Computer Science and Information Engineering
Shu-Te University
No.59, Hengshan Rd., Yanchao, Kaohsiung County 824, Taiwan
s11639113@stu.edu.tw

Wu-Chih Hu
Department of Computer Science and Information Engineering
National Penghu University of Science and Technology
No.300, Liuhe Rd., Magong City, Penghu County 880, Taiwan
wchu@npu.edu.tw

Received Jan. 2018, Revised Feb. 2018

Abstract. *In recent years, the popularity of social media services (such as Facebook, Twitter, and Plurk) has been on the rise. As users grow accustomed to posting their personal opinions on certain issues on the Internet, collecting data on the public's opinion and attitude towards specific issues can provide the government, organizations, and corporations with a solid basis of reference for their decision-making. Thus, sentiment analysis (SA) of text messages has become a crucial research topic, while the value of research results and related application have gained more emphasis and more emphasis. This study proposes constructing an intelligent sentiment analysis classifier by integrating CCLM (Combined CKIP Language Model) with naïve Bayes classifiers, with the aim of establishing a high-accuracy sentiment classification based on Mandarin texts posted on social media platform to provide more accurate predictions towards future public opinion trends and behavior patterns. In our experiment, the used short text messages derived from a famous microblogging service – Plurk – as our subject of study to explore sentiment analysis of Mandarin sentences; the naïve Bayes classifier combined with three language models – BLM (bigram language model), CLM (CKIP language model, and CCLM – to construct sentiment analyzers. Throughout different size of data sets in the experiment, the sentiment classifier adopting CCLM achieved better accuracy.*
Keywords: Naïve Bayes Classifier, Natural Language Processing, Sentiment Analysis, Opinion Mining.

1. **Introduction.** With the emergence of Web 2.0, popularity of handheld smart mobile devices and social media services, a few famous examples being Wikipedia, Blog, Facebook, have risen rapidly; among them, the most representative are microblogs, which provide the social media service of allowing users to post their subjective thoughts or opinions on specific issues online at any place, any time. The remarks in these posts reflect the user's truthful feelings; when they resonate with that of other people's, they receive comments of varying amount and mixed reviews, creating a new form of interpersonal interaction that has rendered these abundant text contents the subject of study for many related issues.

Sentiment analysis [1, 2] [12], also known as opinion mining [3], applies processing technologies including natural language processing (NLP), machine learning, data mining, and information retrieving; the processing operation goes through short messages to determine the polarity [8] (positive, neutral, or negative) of an author's view on a specific issue. The public's sentiments are closely related to the results of many issues or developments. Take elections as an example; the voters' impression or opinions of a certain candidate can indirectly reflect their voting behavior. When surveying public opinion through sentiment analysis for the purposes of making market orientation or public affair decisions, not only should we analyze the sentiment behind words, but we should also take factors such as topic, author, and narratives into consideration in order to fully exploit the benefits of sentiment analysis. In recent years, sentiment analysis researches have been applied by business intelligence towards decision-making on enterprise products as well as prediction of public affair elections or voting results. Eminent sentiment analysis applications include predicting future stock price trends by looking at stock investors' abundant discussion on stock message boards [5], analyzing the public's movie reviews through machine learning to decide whether or not to recommend a movie [4] [6] [21], predicting a restaurant's rating by looking at customers' reviews on the restaurant's food ingredients and dining ambience [7], and, in the field of education, designing an intelligent tutoring system based on observations of the learner's emotional expression changes during their learning process [24].

Plurk is highly representative of Mandarin social media services. This study employed posts on Plurk as the subject of sentiment analysis. Each Plurk post has a time stamp that can be traced, their length is limited to 140 characters, and bear the trait of being subjective views of the author; hence, in terms of emotional expression, they reflect the author's genuine opinions. Meanwhile, given the length restriction, users must be concise in expressing themselves, reducing the possible number of attributes to be retrieved. In sum, the focus of this study lies on discussing how to utilize these sentence-level posts to efficiently and accurately retrieve opinions and evaluate whether they can be applied in predictive analysis.

Term segmentation is the first step in sentence analysis. An appropriate term segmentation can form an ideal corpus and the quality of a corpus is key to determining the effect of sentiment analysis classification. Sentiment analysis under the Mandarin and English languages face different language structures and complex syntaxes. For instance, English terms are segmented by spaces, an attribute that is not applicable towards term segmentation in Mandarin because the terms are clustered together. Our general Mandarin term segmentation analysis was completed under assistance of the Chinese Knowledge Information Processing (CKIP) system, developed by Taiwan's Academia Sinica. Upon completing term segmentation analysis, we proceeded to establishing the language model. When constructing a Mandarin language model, the choice is usually unigram or bigram, both of which are n-gram models [18]; many sentiment analysis studies also employ one of these two models or a combination of both to improve the accuracy of Mandarin sentiment classification [22]. This study uses the bigram technique and combines, in order, segmentation yielded by CKIP analysis of Mandarin sentences to propose a new language model, CCLM (Combined CKIP Language Model). Furthermore, we combine the above with naïve Bayes classifiers to design a brand new sentiment analyzer that accomplishes the goal of enhancing positive/negative sentiment classification of Mandarin texts.

2. Related Works.

2.1. Sentiment Analysis.

Generally speaking, sentiment analysis processing involves two parts. The first part is extracting words in an article that express opinion. Based on the length of target words, sentiment analysis can be divided into three levels: document level, sentence level, and phrase level; different levels call for different analysis techniques. For instance, Gamon [9] proposes two different processes for sentiment analysis at the document level and at the sentence level, retrieving sentence-level articles for analysis of positive/negative comments and then yielding conclusions from such results [10].

The second part involves analyzing attitudes, which can commonly be classified as positive, negative, or neutral. Emotional states can be further classified into anger, surprise, sadness, disgust, fear, and joy [11]; based on text dependence, factors including time, context, social community, and text replies are all taken into consideration for emotional state determination so as to improve sentiment classification accuracy. We found that the relationship between individuals in a social community can easily encourage shared emotion or similar opinions and views with respect to a topic [13]. Given the above, we added replies within a community and text sentiment attributes created on the same timeline to increase the accuracy of text sentiment classification. Hence, applying social networking analysis towards sentiment analysis can indeed yield satisfactory results [23].

2.2. Term Frequency.

When constructing a language model and its corpus, we compile statistics on each term's occurrence frequency. Term frequency (TF) [14, 15] is used to calculate a given term's occurrence frequency across all texts. Generally speaking, a term's significance is directly proportional to the number of times it occurs. However, some terms might occur frequently but bear little significance; such terms include function words such as conjunctions, and pronouns (such as between, at, and of), and stop words (such as a, as, and be). These terms are first to be ignored or filtered out when processing terms in documents. Term frequency (TF) is calculated as shown in Formula (1):

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

Where $n_{i,j}$ is the number of occurrences of the considered term t_i in document d_j, and the denominator is the sum of number of occurrences of all terms (t_1,...,t_k) in document d_j. Considering the frequency of which term n_i occurs in classification c, Formula (1) can be represented as the following in Formula (2):

$$\hat{P}\left(n_i|c\right) = \frac{\sum_{i\in c} n_{i,j}}{\sum_{k\in c} n_{k,j}} \tag{2}$$

In the formula, $\hat{P}\left(n_i|c\right)$ stands for the occurrence frequency of term n_i under classification c; $\sum_{k\in c} n_{k,j}$ stands for the total number of terms (including repeated ones) under classification c; $\sum_{i\in c} n_{i,j}$ stands for the number of occurrences of term n_i in document d_j.

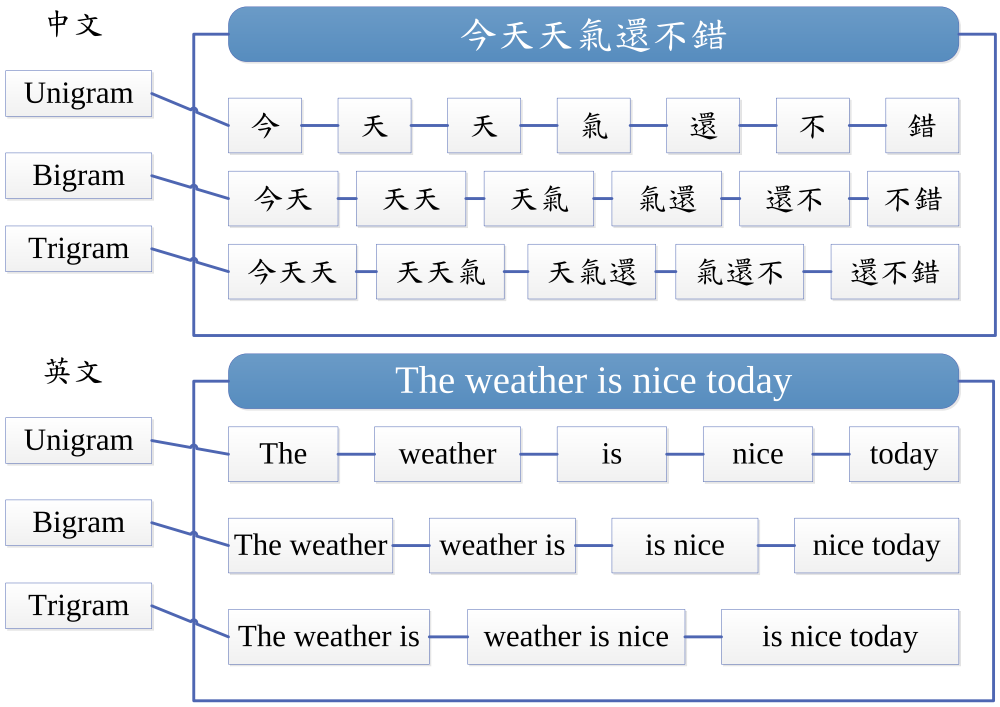

FIGURE 1. Illustration of n-gram model application on Mandarin and English texts.

2.3. **N-gram.** In sentiment classification processing, language models play quite an important role; among different models, n-gram models [16, 17] are the most widely used for language and statistical processing. N-gram is a sequence of n items from a given sequence of text. Prior to constructing a model, we first select the model basis from unigram, bigram, trigram... up until n-gram that corresponds to n=1, 2, 3,..., n. We then take the training dataset, composed of collected texts, and run them through the n-gram model to obtain all the term sequences in each text, and add them into the established corpus. Different languages yield different effects under different n-gram models. As shown in Figure 1, in Mandarin, a meaningful term is usually composed of two or more characters; randomly segmenting two characters into a term might yield a meaningless term. This is because it is harder to determine term segmentation in Mandarin texts; by contrast, English terms are separated by spaces, so each term is able to retain its original meaning even after recombination. Hence, applying n-gram to English term segmentation analysis yields better results than to Mandarin.

Bayes classifiers is based on Bayes Theorem [19]; they rely on probabilistic and supervised-learning approaches that employ sample training and constant learning to achieve effective classification. Bayes classifiers can be used in sentiment classification of opinions and comments in documents; the supervised statistics method renders successful statistical analysis that helps classification of positive/negative sentiments [20].

The relation between documents and document classification can be interpreted as: when given conditions of document d, the probability of the document belonging to classification c can be calculate as follows, in Formula (3):

$$P(c|d) = \frac{P(c \cap d)}{P(d)} = \frac{P(d|c)\,P(c)}{P(d)} \tag{3}$$

$P(c|d)$ is the prior probability, which represents the distribution probability of classification c based on prior observations; c stands for classified items while $P(c|d)$ stands for the posteriori probability, representing the probability of being distributed to classification c after document d's attribute training. The Maximum-a-Posteriori probability is expressed as follows, in Formula (4):

$$c_{map} = \underset{c \in C}{argmax}\, P(d|c)\, P(c) \tag{4}$$

Since all of document d's classification similarity share the same denominator, we can omit it and assume that document d's attribute term distribution $(t_1, t_2, \ldots, t_i)$ are independent of each other, as shown in the following Formula (5):

$$\begin{aligned} & c_{map} = \underset{c \in C}{argmax}\, P(t_1, t_2, ..., t_n|c)\, P(c) \\ & = \underset{c \in C}{argmax}\, P(t_1|c) * P(t_2|c) * ... * P(t_n|c) * P(c) \\ & = \underset{c \in C}{argmax}\, P(c) \prod_{t=1}^{n} P(t_n|c) \end{aligned} \tag{5}$$

Since we are assuming that attribute samples are independent of each other, we can use the known probability model to estimate the classification category similar to that of the unknown data's; the more similar they are, the greater the similarity is. The greatest advantage of employing Bayes classifiers is when new attribute samples are added in, it only requires minimal adjustment of the probability distribution. Hence, the approach bears several features of being simple in operation and high in efficiency when applied to classification.

This study improves the accuracy of sentiment classification by integrating the new CCLM language model and Bayes Theorem. The proposed approach first applies CKIP to complete term segmentation of Mandarin short texts collected from social media platform; we then take the bigram (an n-gram in which n=2) collected from such results and construct a new sentiment analysis language model, CCLM. Following that, we use the corpus from CCLM as naïve Bayes classifiers to establish a sentiment analyzer. The research process of improving Mandarin sentiment classification can be divided into four steps, as illustrated in Figure 2. In the first step, we collect Mandarin text data from Plurk via web agents and proceed to preprocess the collected short text messages. Afterwards, we utilize model training to finish constructing a Bayes sentiment classifier for Mandarin texts. Lastly, we use the trained Bayes sentiment classifier to finish subsequent Mandarin text sentiment classification.

2.4. **DATA COLLECTION.** Compared to Twitter and Facebook, two mainstream social media services of which texts are mostly in English, posts on Plurk are mostly in Mandarin. Therefore, we chose Plurk as the source of document collection for our study's training dataset. The retrieval process of the whole document is an automated collection of target documents by web agents through Plurk Search. Prior to conducting training on the sentiment dataset, we must first label the newly added document's sentiment classification polarity; however, given Plurk's large amount of documents, to manually label sentiment classification would be very time-consuming. On the other hand, if we can make use of the documents' emoticons (as shown in Figure 3) to assist in automatically determining a document's polarity, then we may save a lot of time. Among all the widely used emoticons on Plurk, this study only employs the symbols that most frequently display positive/negative emotions – ":-D", which represents positive emotion, and ":-(", which represents negative emotion – in the collection process.

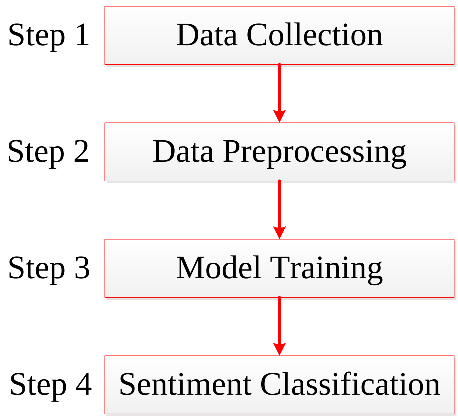

FIGURE 2. Processing steps of the proposed method.

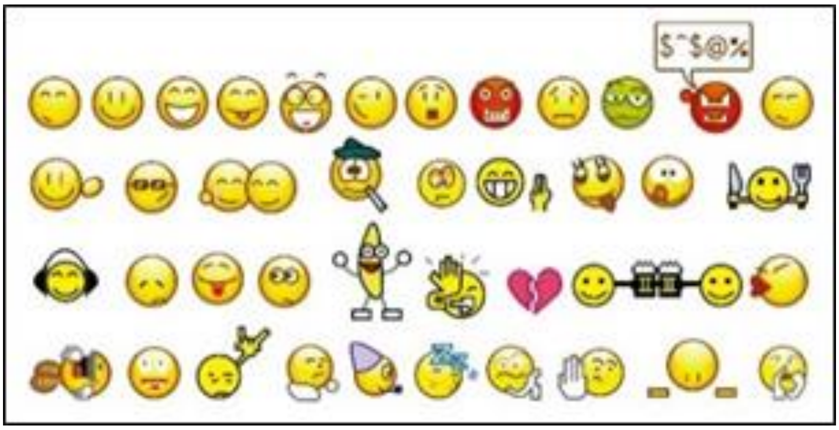

FIGURE 3. Examples of Emoticons.

2.5. **Data Prepocessing.** Although using emoticons for determination can help save time in classification, occurrences of exceptions might lead to misjudgment, so we added a layer of data preprocessing to weed out noise and avoid misjudgment. Overall, data preprocessing can be divided into two steps. The first step is finding the sentence that corresponds to the emoticon. A document might include more than one emoticon, and each emoticon might correspond to a different sentence. Take a look at the following example: ":-(I'm so hungry; let's all go to McDonald's! :-D" The sentence "I'm so hungry" corresponds to the emotion of ":-(" while "let's all go to McDonald's" corresponds to the emotion of ":-)". Thus, we must first figure out the each emoticon's relative position in its corresponding sentence, which can be one of the following four positions: left, center, right, and all, as shown in Figure 4. Upon gathering the statistics, we have observed that, in a sentence, emoticons mostly occur on the right (66%) and center (29%); therefore, we first filter out any document that doesn't belong to either of these two while retaining cases in which the attributes is right or center. If a case has emoticons in the center but includes multiple emoticons, then we segment the case based on each emoticon and have it correspond to the sentence on its left. The second step concerns further filtering of sentences that have already been given labels based on their emoticons. Since collected sentences might still include different emoticons, a second-time filtering is needed to remove non-selected emoticons and place them in the corpus.

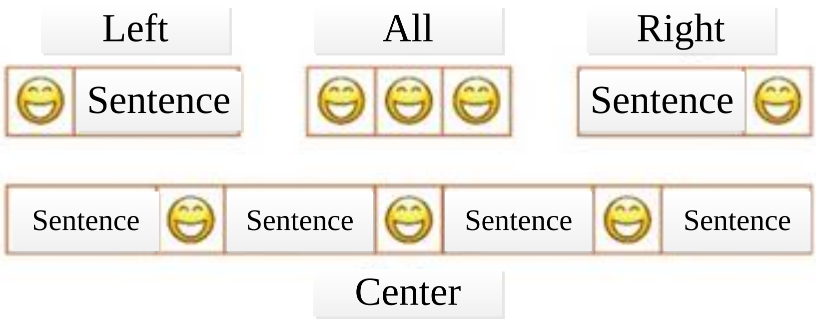

FIGURE 4. Examples of Corresponding Position of Emoticons in Sentences.

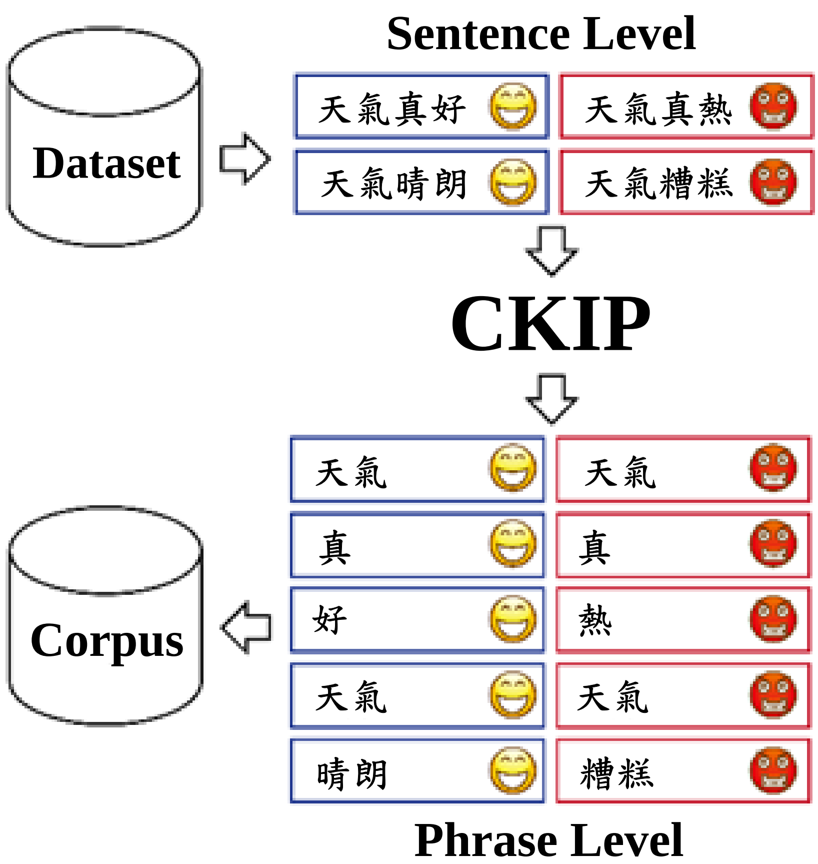

FIGURE 5. Examples of Mandarin Term Segmentation of the Dataset through CKIP.

2.6. **Model Training.** Model training procedure applies CKIP on sentences that have undergone data preprocessing to accomplish Mandarin term segmentation. The output, corresponding language models are obtained by taking the sentences that have completed term segmentation, splitting them into phrases, and then constructing three language models: BLM (bigram language model), CLM (CKIP language model), and CCLM. They are labeled with the original sentence's emotion, as shown in Figure 5.

After retrieving the sentence "天氣真好," we transmit it via CKIP's APID method to the CKIP server for term segmentation. After the sentence is split into the phrases "天氣," "真," and "好," the result is sent back to the local end; each phrase is then labeled with the corresponding emotion of the original sentence and, finally, all the phrases are placed

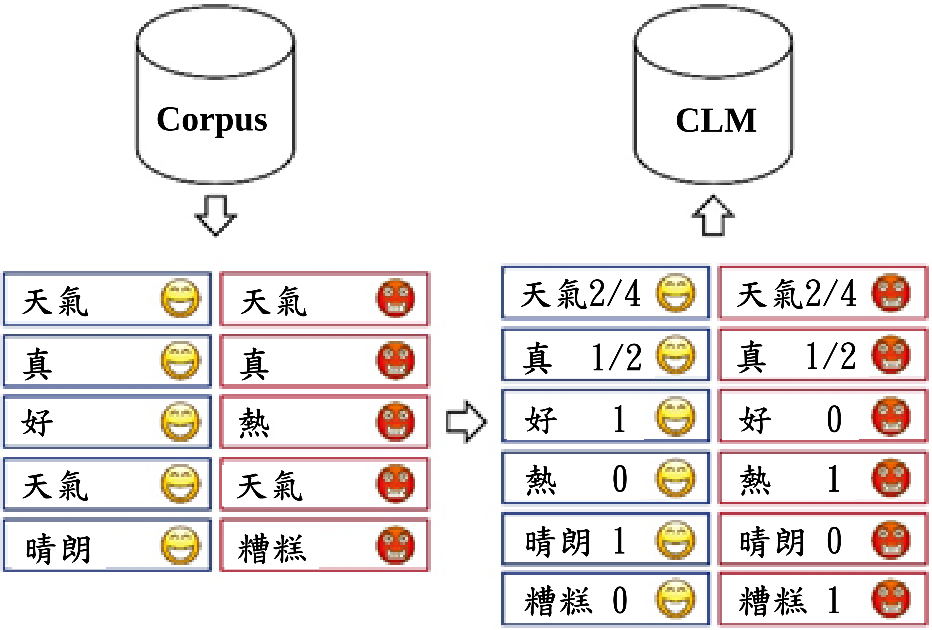

Figure 6. Example of Constructing a CLM Language Model.

in the corpus. After storing all the phrases that have undergone term segmentation into the corpus, the corpus is then used to establish the CLM model and compile the frequency at which corpus phrases and emotions occur, as shown in

Figure 6. Take the four times ”天氣” occurred in the corpus as an example to illustrate the processing procedure: of the four times the phrase occurred, two were labeled positive and two were labeled negative. In the compilation process, the order is to compile phrases first and then emotions. Compilation results show that for the phrase ”天氣,” positive emotions account for 1/2 of the frequency; negative emotions also account for 1/2 of the frequency; after calculating the frequency of these emotion phrases, results are recorded in CLM.

We can see that after CLM splits the sentences into phrases, some meanings change from the original. For instance, it the input sentence is ”我不喜歡讀書,” then CKIP term segmentation yields ”我,” ”不,” ”喜歡,” and ”讀書,” which changes the meaning of 不喜歡. To resolve such issues, this study proposes the CCLM model, which combines CKIP term segmentation with bigrams. CCLM is established by first retrieving each phrase from the corpus, and then coupling the phrases through the form of bigrams, as illustrated in Figure 7.

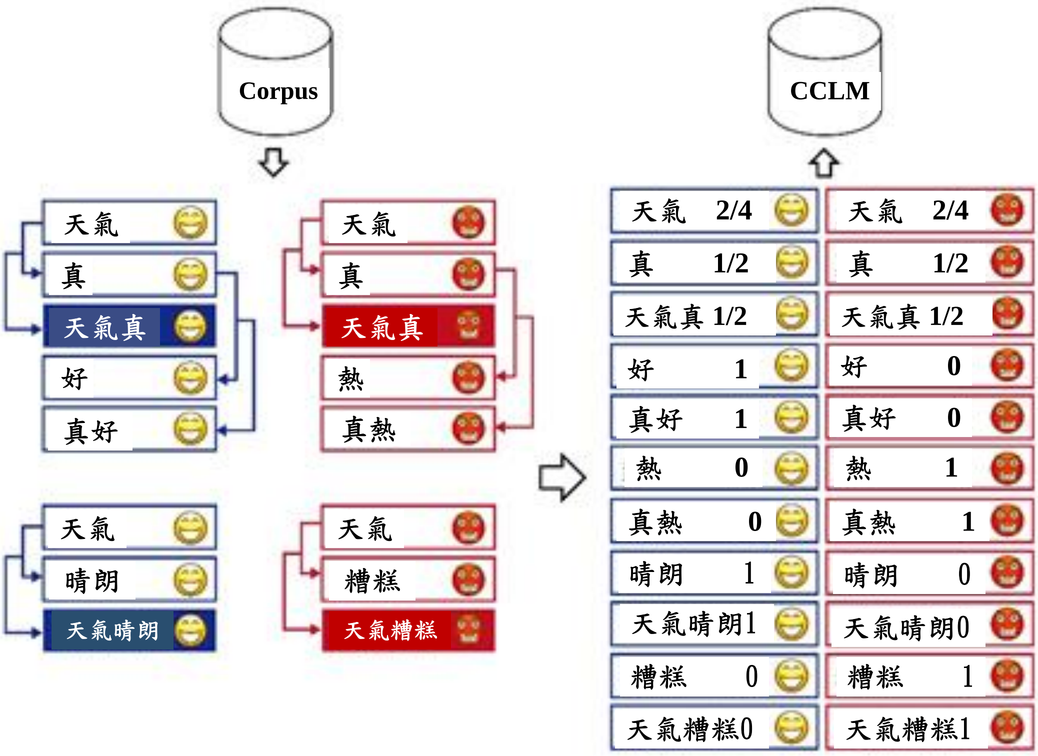

FIGURE 7. Example of Coupling Phrases through Combination of CKIP Term Segmentation and Bigrams.

2.7. **Sentiment Classification.** This study applies a combination of naïve Bayes classifiers with CCLM to conduct sentiment classification, thus improving the calculation method of Bayes attributes. As illustrated in Formula (6):

$$C_{NB} = \underset{C_S}{argmax}\, P\left(C_S\right) * \prod\nolimits_{t \in X} P\left(X|C_S\right) \tag{6}$$

Suppose there is a document d; to retrieve phrase attributes, we use CKIP for term segmentation, and the resulting phrase attributes are represented as set Y. Phrases in set Y are then combined in a bigram form, and the resulting phrase attributes are represented as set X. With C_S standing for positive/negative sentiment classification, $P\left(X|C_S\right)$ calculates phrase attributes, as illustrated in Formula (7). The purpose of $\frac{1}{|V|}$ is to provide Laplace smoothing in order to prevent instances in which, when a certain phrase does not occur in the language model, given conditional independence, the denominator risks being 0. $|V|$ is the total number of phrases; count(c) is the number of phrases under classification C; count(X, C) is the number of times X occurs under classification C.

$$P\left(X, C\right) = \frac{count\left(X, C\right) + 1}{count\left(C\right) + |V|} \tag{7}$$

Where $P(X|C_S)$ represents the probability of phrase x occurring under sentiment classification condition of C_S. When phrase x displays occurrence of 0 times in CCLM corpus, it means that there exists no such phrase, and we; by contrast, when phrase X occurs more than 0 times, we remove all phrases containing y, eventually using $X \cup Y$ to reconstruct the final corpus, set T.

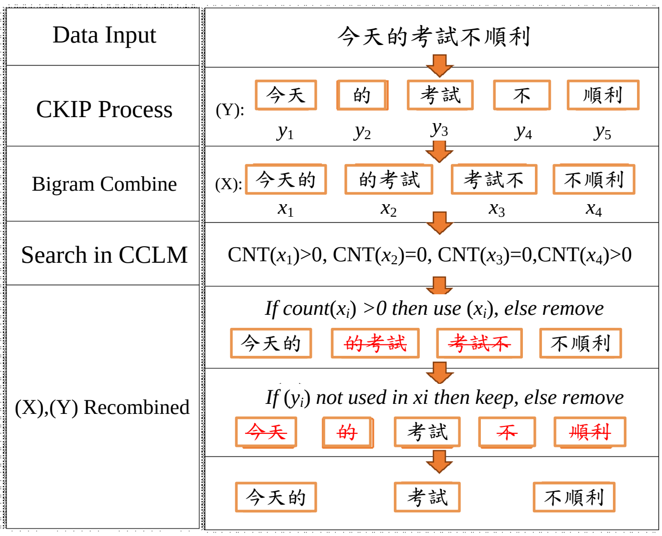

FIGURE 8. Preprocessing Example of CCLM Document Classification Phrase Attributes.

Take Figure 8 as an example. We have a document, "今天的考試不順利", that awaits sentiment classification. We first run it through CKIP for term segmentation to split it into phrases, which are then combined using bigrams. The following step is to determine whether such combinations have occurred in CCLM; if not, then the combination is replaced by the original, individual phrases. Finally, we input the newly combined attribute phrases to calculate C_{NB}, and classify document d with the classification that yielded the highest C_{NB}.

3. **Results.** Considering the processing time of language models BLM, CLM, and CCLM, the experiment implementation's hardware environment constitutes the following: the processor was Intel(R) Xeon(R) CPU E31220 @ 3.10GHz 3.10GHz, the memory size was 8GB, and the hard drive size was 500GB. Meanwhile, for the software environment, we established our web server through Apache and our database using MySQL while opting for the programming language PHP to realize all the experiment steps.

As for the document source of our study's training dataset, it was obtained by using the two emoticons ":-D" and ":-(", respectively representing positive and negative emotions, as keywords through Plurk Search via web agents. We obtained a total of 65,368 Plurk documents that were published between 2008, when Plurk became open to public use, and May 2014. Positive documents account for 32,855 pieces while negative documents account for 32,513. We collected an extra 1,000 test documents that were manually labeled for classification determination in order to test the classifier's accuracy.

Given that the document containing collected corresponding emoticons might contain other mixed emoticons, a second-time filtering is necessary. After removing documents with non-selected emoticons, the number of documents drops from 65,368 to 52,694 sentences, and the data is stored in the training dataset. We took the 52,694 sentences in the training dataset and split them up, applying a test unit of increasing 10,000 sentences each time to observe, when given different number of training sentences, the statistical results of each language model's phrase count, execution efficiency, and accuracy. We can see from Figure 9 the phrase count of language models BLM, CLM, and CCLM; all models present a linear growth for different training data amount, with CCLM showing the greatest growth margin, BLM taking second place, and CLM having the least growth.

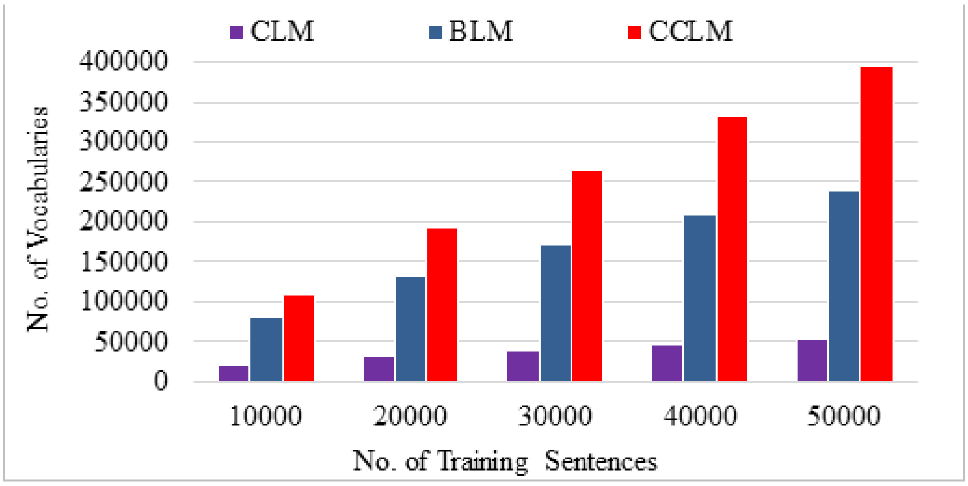

Figure 9. Phrase Count Comparison of CLM and CCLM.

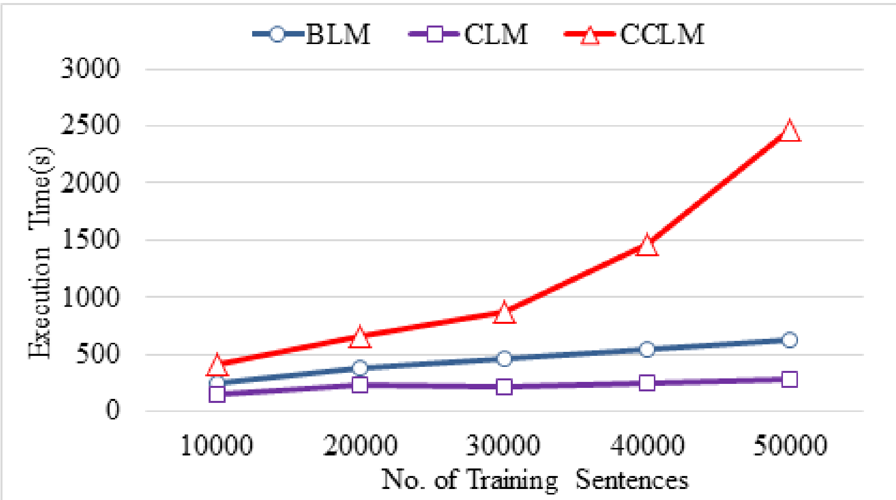

Figure 10. Comparison of Execution Time Between CLM and CCLM.

When 50,000 training sentences are given, the difference between CLM and CCLM comes to 340,000 phrases.

When it comes to overall execution efficiency, CCLM takes the longest to process; BLM and CLM come in second and third respectively. From our observation, the execution time length is proportional to statistical results of phrase count. It can be inferred that when there are more phrases, the overall execution efficiency is affected; moreover, if the number of training sentences continues to increase, the execution time might present exponential growth. By contrast, CLM has the lowest number of phrases, and therefore is fastest in operation time; its comparison against the other two models in terms of execution time length growth is also less significant. The growth of BLM's phrase count, as is of its execution time, falls between CCLM and CLM, as illustrated in Figure 10.

From our observation of accuracy in BLM, CLM, and CCLM models, when the training dataset bears fewer sentences, employing BLM yields satisfactory accuracy. However, as the number of training sentences increases, accuracy across all models decreases. From our observation, a possible explanation is that there might exist significant amount of noise data that affects accuracy. Nevertheless, as data increases to 20,000 sentences, CCLM

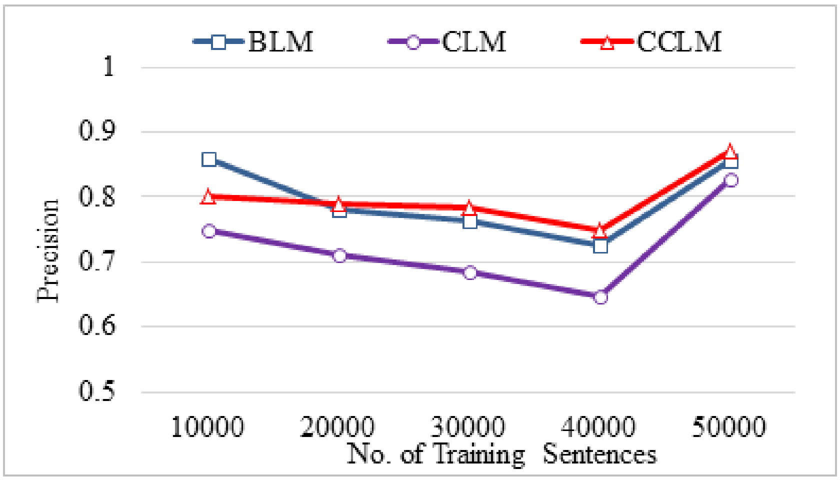

FIGURE 11. Comparison Results of Accuracy of BLM, CLM, and CCLM.

displays better accuracy than both CLM and BLM; at one point, CLM's accuracy fell behind, and only rose to 80% after reaching 50,000 sentences, which is also when CLM and CCLM reached their peak accuracy. As seen in Figure 11, we can thus conclude the following: we should select the suitable language model when given distinct conditions; for instance, CLM works well in cases that require fast processing but are lenient towards accuracy, BLM offers decent balance between processing time and accuracy without sacrificing either, and CCLM emphasizes high accuracy at the price of longer processing time.

4. **Conclusion.** Sentiment analysis has been prominent in English for many years now; related analytic products have also been popular. This study proposes a method to improve sentiment classification of Mandarin short text messages on Chinese social media services. All Mandarin short text messages were collected from Plurk as the source of training dataset, and conduct training to have a combination of CCLM language model and Bayes classifiers that improves accuracy in sentiment classification. Through the above, the research established a positive/negative sentiment classification system for Mandarin short texts; this system shall allow subsequent sentiment analyses to progress forward.

In order to prove that CCLM does indeed increase sentiment classification accuracy through the experiment, the basis models BLM, CLM, and CCLM, which were then combined with naïve Bayes classifiers to produce a sentiment classifier. Finally, we examined experiment results to observe and test from three aspects – phrase count, execution efficiency, and accuracy – how datasets were making use of sentiment classification systems. Experiment results prove that CCLM does indeed present high sentiment classification accuracy than BLM and CLM; however, because CCLM yields more phrases in the process, it also requires more storage space and computation time.

REFERENCES

[1] B. Pang, L. Lee, and S.P. Mishra, *Opinion mining and sentiment analysis*, Foundations and Trends in Information Retrieval, Vol. 2, No. 1-2, pp. 1-135, 2008.

[2] B. Pang and L. Lee, *A sentimental Education: Sentiment Analysis using Subjectivity Summarization Based on Minimum Cuts*, in Proceedings of 42nd Annual Meeting on Association for Computational Linguistics, Barcelona, Spain, 2004.

[3] A. Esuli and F. Sebastiani, *Sentiwordnet: A Publicly Available Lexical Resource for Opinion Mining*, in Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC2006), Genoa, Italy, 2006.

[4] P. Turney, *Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews*, in Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Sofia, Bulgaria, 2002.

[5] S. Das and M. Chen, *Yahoo! for Amazon: Extracting market sentiment from stock message boards*, in Proceedings of the Asia Pacific Finance Association Annual Conference, Bangkok, Thailand, 2001.

[6] B. Pang, L. Lee, and S. Vaithyanathan, *Thumbs up?: sentiment classification using machine learning techniques*, in Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, New Jersey, USA, 2002.

[7] B. Snyder and R. Barzilay, *Multiple aspect ranking using the good grief algorithm*, in Proceedings of the Human Language Technology/North American Chapter of the ACL Conference, New York, USA, 2007.

[8] S.M. Kim and E. Hovy, *Determining the Sentiment of Opinions*, in Proceedings of the 20th International Conference on Computational Linguistics, Geneva, Switzerland, 2004.

[9] M. Gamon and A. Aue, *Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms*, in Proceedings of the 43th Annual Meeting on Association for Computational Linguistics, Ann Arbor, USA, 2005.

[10] M. Hu and B. Liu, *Mining Opinion Features in Customer Reviews*, in Proceedings of the 19th National Conference on Artificial Intellgience, San Jose, USA, 2004.

[11] P. Ekman, *An argument for basic emotions*, Cognition & emotion Vol. 6, No. 3-4, pp, 169-200, 1992.

[12] P. Routray, C.K. Swain, and S.P. Mishra, *A Survey on Sentiment Analysis*, International Journal of Computer Applications, Vol. 76, No. 10, pp. 1-8, 2013.

[13] C. Tan, L. Lee, J. Tang, L. Jiang, and M. Zhou, P. Li, *User-level Sentiment Analysis Incorporating Social Networks*, in Proceedings of the 17th International Conference on Knowledge Discovery and Data Mining, NY, USA, 2011.

[14] H.C. Wu, R.W.P. Luk, K.F. Wong, and K.L. Kwok, *Interpreting tf-idf Term Weights as Making Relevance Decisions*, ACM Transactions on Information Systems, Vol. 26, No.3, pp.1-37, 2008.

[15] G. Salton, A. Wong, and C.S. Yang, *A Vector Space Model for Automatic Indexing*, Communications of the ACM, Vol. 18, No. 11, pp. 613-620, 1975.

[16] P.F. Brown and P.V. Desouza, *Class-Based N-gram Models of Natural Language. Computational linguistics*, Computational Linguistics, Vol. 18, No. 4, pp. 467-479, 1990.

[17] W.B. Cavnar and J.M. Trenkle, *N-gram-based text categorization*, in Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, Nevada, 1994.

[18] Claude E. Shannon and Warren Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, 1949.

[19] A. Park and P. Paroubek, *Twitter as a Corpus for Sentiment Analysis and Opinion Mining*, in Proceedings of the 7th International Conference on Language Resources and Evaluation, Valletta, Malta, 2010.

[20] H. Yu and V. Hatzivassiloglou, *Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences*, in Proceedings of the 41st Empirical Methods in Natural Language Processing, Sapporo, Japan, 2003.

[21] P. Chaovalit and L. Zhou, *Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches*, in Proceedings of the 38th International Conference on Hawaii International Conference on System Sciences, Hawaii, USA, 2005.

[22] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin, *Learning sentiment-specific word embedding for twitter sentiment classification*, in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Vol. 1, pp. 1555-1565, 2014.

[23] W. Gryc and K. Moilanen, *Leveraging textual sentiment analysis with social network modelling*, From Text to Political Positions: Text analysis across disciplines, Vol. 55, pp. 47, 2014.

[24] K.L. Hao-Chiang, W.A.N.G. Cheng-Hung, C.H.A.O. Ching-Ju, and M.K. CHIEN, *Employing Textual and Facial Emotion Recognition to Design an Affective Tutoring System*, The Turkish Online Journal of Educational Technology, Vol. 11, Issue 4, pp. 418-426, 2012.