Deep Learning-Based Idiomatic Expression Recognition for the Amharic Language

Demeke Endalie (demeke.endalie@ju.edu.et)
Jimma institute of technology
Getamesay Haile
Jimma institute of technology
wondmagegn taye
Jimma institute of technology

Research Article

Keywords: Amharic language, CNN, FastText, idiomatic expression, idiomatic recognition

Posted Date: August 2nd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1901631/v1

Abstract

Idioms are used in Amharic to conceal information or to express ideas indirectly. However, most natural language processing models used with the Amharic language, such as machine translation, semantic analysis, sentiment analysis, information retrieval, question answering, and next word prediction, do not consider idiomatic expressions. As a result, in this paper, we proposed a conventional neural network (CNN) with a FastText embedding model for detecting idioms in an Amharic text. We collected 1700 idiomatic and 1600 non-idiomatic clause datasets from Amharic books to test the proposed model's performance. The proposed model is then evaluated using this dataset. With testing and training datasets, the proposed model achieves an accuracy of 80% and 98%, respectively. We compared the proposed model to other machine learning models like K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Random Forest classifiers. According to the experimental results, the proposed model produces promising results.

1. Introduction

In recent years, the development of deep learning in neural networks improves performance in many natural language processing (NLP) tasks. In natural language processing, neural networks are used for the development of machine translation, speech recognition, text generation, text mining, and named entity recognition.

Idiomatic expression is a collection of words that have a different meaning from the individual words in them. The meaning of the idioms cannot be interpreted from the meaning of words that constructs them directly [1]. Idiomatic expressions are one of the important parts of all-natural languages [2]. The detection of this type of expression from Amharic text helps those individuals that are not familiar with the Language. For example, the expression “□□□ □□□” can be directly translated as “he drops his face” but the actual meaning is “he becomes sad”.

Idiomatic expression recognition from a given text plays an important role in the implementation of tasks such as machine translation, speech recognition, sentiment analysis, and dialog system within the respective language. Amharic is one of the languages grouped under the Semitic language families that have more than 4000 idiomatic expressions [3].

The paper by [4] presents the use of Skip-Thought Vectors to create distributed representations that encode features that are predictive concerning idiom token classification. They showed that classifiers using these representations have competitive performance compared with the state of the art in idiom token classification. However, their models use only the sentence containing the target phrase as input and are thus less dependent on a potentially inaccurate or incomplete model of discourse context. They further demonstrate the feasibility of using these representations to train a competitive general idiom token classifier.

The authors of [5] proposed an idiomatic expression detection method based on the assumption that idioms and their literal counterparts do not occur in the same contexts. The inner product of context word vectors with the vector representing a target expression is computed first by their model. Because literal vectors predict local contexts well, their inner product with contexts should be greater than idiomatic ones. This distinguishes literals from idioms, and then in word vector space, computes literal and idiomatic scatter (covariance) matrices from local contexts. Because the scatter matrices represent context distributions, they used the Frobenius norm to calculate the difference between the distributions.

The work of [6] presents a generalized model for determining whether an idiom is used figuratively or literally, based on the concept of semantic compatibility. They examine continuous bag-of-words (CBOW's) limitations in terms of semantic compatibility measurement and propose a novel semantic compatibility model based on CBOW training for idiom usage recognition. Experiments on two benchmark idiom usage corpora reveal that the proposed generalized model outperforms state-of-the-art per-idiom models at the time.

The Authors of [7] offer a model for detecting idiomatic phrases in written text. They attempted to recognize idioms as an anomaly and supervised sentence categorization. For outlier detection, they use principal component analysis. Idiom detection as lexical outliers does not make use of class label information. As a result, in their experiments, the authors utilize linear discriminant analysis to generate a discriminant subspace and then use the three nearest neighbor classifiers to calculate accuracy. They analyze the advantages and disadvantages of each technique. All of the techniques are broader than earlier idiom identification algorithms in that they do not rely on target idiom types, lexicons, or huge manually annotated corpora, nor do they confine the search area to a certain form of linguistic construction.

Idiomatic expression in language has a detrimental impact on NLP task performance [8]. However, according to the researchers' understanding, there is no Amharic natural language processing model that considers idiomatic expression. This inspired us to create an Amharic idiomatic phrase identification system based on deep learning. This study focuses on the construction of a CNN using the FastText model to detect the presence of idiomatic terms in an Amharic text. The overall contributions of the study are summarized as follows:

1. Prepare a general-purpose Amharic idiomatic expression dataset that can be used by other studies in the future.
2. Proposed a deep learning model that incorporates CNN with FastText to recognize idioms from Amharic texts.
3. Evaluate the performance of the proposed recognition model with various evaluation metrics.

The remainder of the paper is structured as follows. Section 2 presents the planned work's comprehensive methodology in detail. Section 3 defines the experimental results. In section 4, we present the outcome and a discussion of it. Finally, section 5 is the conclusion.

2. Materials And Methods

This study focuses on the development of a deep learning model using FastText to detect the presence of idiomatic terms in an Amharic text. Figure 1 below depicts the proposed idiomatic expression recognition architecture for the Amharic language. Pre-processing, word embedding, and learning modules are all components of the proposed automatic idiomatic expression identification system. The tasks in the proposed model range from data gathering to evaluation. This means the proposed model contains tasks from data collection up to evaluation.

2.1. Dataset

The dataset utilized in this study was gathered from two Amharic books "□□□□□ □□□□" (idiomatic expressions in Amharic), and "□□□ □□□ □□□□" (love up to the grave) [3, 9]. Most idiomatic expressions in books are 2 to 4 in number of tokens, so the dataset contains only 2 to 4 length idiomatic expressions. There are more than four thousand idioms in the Amharic writing system. We received 1700 idiomatic words from the aforementioned books, all of which are easily readable in the books themselves. In addition to idioms, we also collect phrases that are not classified as idiomatic expressions in the book to train the proposed model. After collecting the data from books, we apply the following preprocessing modules to clean up the data and make the learning phase as easy as possible.

i. Normalization

The Amharic writing system has different letters ("□□□□") that can be read with the same pronunciation, but there are no rules to distinguish their meanings. As a result, in Amharic, the same concept or name of an object may be represented by these letters. This increases the number of features extracted for processing or analysis. To overcome this redundancy, we normalize those characters with the same pronunciation to one canonical letter used in this study, as shown in Table 1 below [10].

Table 1
Normalization of characters having the same pronunciations.

Canonical character	Characters with the same pronunciation as the canonical character
□(hā)	□□□□□□□□□(hā)
□(še)	□(še)
□(ā)	□□0□□(ā)
□(ts'e)	□(ts'e)
□(wu)	□(wu)

ii. Stemming

Stemming is the process of reducing inflected words to their stem, base, or root form. Amharic is one of the morphological-rich Semitic languages [11]. Different terms can exist with the same stem, and this

helps reduce the size of feature space for processing. In this study, we used the HornMorpho stemmer developed by Michel Gasser [12]. HornMorpho is a Python library that is developed for the analysis of three Ethiopian languages Amharic, Afan Oromo, and Tigrigna.

iii. Remove stop words

In Amharic, the common words, e.g. "□□□ □□□□ □□□□□" and others that scoreless weightage in the text processing tasks is called stop words. Stop words are eliminated to save computational time wasted in processing them. Amharic does not have a well-prepared list of stop words. However, we eradicate stop words by [10]. In addition, to stop word removal, we also replace numbers with their name in alphabetic characters ("□□□"). For example, in "2 □□□", 2 can be changed to two ("□□□") and produce "□□□ □□□". This replacement is done by keeping a map of key-value relation between digits and an alphabetic description of each digit.

2.2. Text representation

Encoding is highly required to pass texts as input to different machine learning and deep learning models [13]. One of the text encoding algorithms that changes a given text into a vector is the word2vec algorithm. It is a set of neural network models used to represent a word in a vector space. Those words which have similarities in their context are clustered together and those that do not have any contextual meaning similarity appear sparsely on the vector space. However, word2vec fails to generate the vector of words that are not in the training vocabulary.

FastText is one of the state-of-the-art word embedding models developed by Facebook. For 157 languages, Facebook develops pre-trained FastText embedding models. One of the languages with a trained FastText word embedding model is Amharic. FastText embedding's strength is that it can create a vector for a given term even if it is not in the training vocabulary. This is resolved by taking into account the character-level n-gram of a given term. Because of this, we created a vector of both idiomatic and non-idiomatic (literal) words to train and test the suggested detection model using the pre-trained FastText word embedding. The pre-trained FastText model is used to build the vector for each word in the example algorithm 1 below.

Algorithm 1: word vector generation

Input: FastText model and dataset

Out: Vector

Begin:

model = load_facebook_vectors("cc.am.300.bin")

for every $idiom_{(j)}$ in the corpus

for every $token_{(i)}$ in $idiom_{(j)}$

$vector_{j,i}$ = model.get_vector($token_{(i)}$)

endfor

endfor

end

2.3. Learning Model

2.3.1. Convolutional Neural Network

We need a learning model to determine whether a particular phrase is idiomatic or not. A convolutional neural network is an advanced neural network model that is used to discover patterns and relationships between data items based on their relative positions [11]. CNN can automatically learn effective text feature representation from massive text using a 1D structure (word order) in the convolutional layer. It captures local relationships among the neighbor words in terms of context windows, and by using pooling layers, it extracts global features. CNN is a neural network made up of several convolutional and pooling layers.

2.3.2. K-Nearest Neighbor Classifier

K-Nearest-Neighbors is a basic yet effective non-parametric supervised classification technique. The KNN classifier is the most common pattern recognition classifier because of its effective performance, efficient outputs, and simplicity. It is frequently utilized in pattern recognition, machine learning, text classification, data mining, object identification, and a variety of other domains [14]. The KNN method classifies by analogy, which means that it compares the unknown data point to the training data points to which it is comparable. The Euclidean distance is used to calculate similarity. The attribute values are adjusted to avoid bigger range characteristics from outweighing smaller range ones. In KNN classification, the unknown pattern is assigned the most predominant class amongst the classes of its nearest neighbors. In the event of a tie between two classes for the pattern, the class that has the minimum average distance to the unknown pattern is assigned. Through the combination of several local distance functions based on individual attributes, a global distance function based on distance can be calculated [15].

2.3.3. Support Vector Machine

Support Vector Machines and Kernel methods have found a natural and effective coexistence since their introduction in the early 90s. SVMs use kernels for learning linear predictors in high-dimensional feature spaces [16]. The objective of the SVM algorithm is to find a hyperplane in N-dimensional space (N is the number of features) that distinctly classifies the data points. Hyper planes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyper-plane depends upon the number of features. If the number of input features is 2, then the hyper-plane is just a line. If the number of input features is 3, then the hyper-plane becomes a two-dimensional plane. Figure 2 shows a sample decision boundary separation.

2.3.4. Random Forest Classifiers

A random forest is a technique used in modeling predictions and behavior analysis and is built on decision trees. It contains many decision trees, each representing a distinct instance of the classification of data input into the random forest. The random forest technique considers the instances individually, taking the one with the majority of votes as the selected prediction [17]. A random forest generates a set of decision trees. To achieve diversity among basic decision trees, random forest chose the randomization approach, which works well with bagging or random subspace methods [18]. To generate each tree in the random forest, the following steps should be followed: If the number of records in the training set is N, N records are randomly sampled but replaced by the original data. This is a bootstrap sample. This sample will be a training set for growing the tree. If there are M input variables, a number m $<<$ M is selected, and at each node, m variables are randomly selected from M, and the best split over these m attributes is used to split the node. The value of m remains constant during forest growth. Each tree is cultivated to the best of its ability.

2.4. Evaluation Metrics

The accuracy, precision, recall, and f1-score are used to assess the performance of the models used in this study. The formulas used to calculate them are shown in Table 2 below.

Table 2
Performance evaluation metrics

Evaluation metric	Formula
Accuracy	$accuracy = \frac{Tp + Tn}{Tp + TN + Fp + Fn}$
Precision	$precision = \frac{Tp}{Tp + Fp}$
Recall	$recall = \frac{Tp}{Tp + Fn}$
F1-score	$f1 - score = \frac{2\,(recall * Precision)}{recall + precision}$

Where Tp denotes true positive, Tn denotes true negative, Fp denotes false positive, and Fn denotes false negative.

3. Results And Discussions

All experiments are carried out in a Windows 10 environment on a machine equipped with a Core i7 processor and 16 GB of RAM. The experimental setups used to develop the proposed Amharic idiomatic expression recognition system are shown in Table 3 below.

3.1. Training and validating the Model

We have divided the data to train and validate its performance with a training test split ratio of 80%, 10%, and 10% for training, validating, and testing the proposed model, respectively. To train the proposed CNN model, we tune the hyperparameters using a grid search strategy. The value of the hyperparameters used in this study is shown in Table 3below.

Table 3
Hyperparameters values of CNN model

Hyperparameters	Values
Embedded dimension	300
Number of filters	265
Batch size	16
Dropout	0.5
Activation	Sigmoid
Optimization	Adam
Epoch	100
Loss	Binary cross entropy

The training accuracy and training loss of the model are then displayed in Figs. 3 and 4 below after the model has been trained using the abovementioned parameters and training dataset. Since the training accuracy grows as the number of epoch increases, the model does a good job of learning from the data. In addition, as the number of epochs rises, the training loss declines. This shows that the model picks up on idiomatic expression features from the training set.

3.2. Testing the model

With the testing dataset and the evaluation metrics listed in Table 2 above, we assess the effectiveness of the proposed Amharic idiomatic expression recognition model. Figure 5 below shows the experimental results of how well the proposed scheme performed in terms of accuracy, precision, recall, and f1-measure.

As shown in Fig. 5 above, the proposed Amharic idiomatic expression recognition system, which makes use of CNN with FastText word embedding, achieved results with accuracy, precision, recall, and f1-score of 80%, 70%, 77.78%, and 73.68%, respectively.

3.3. Comparison of the performance of the model with other models

We must take into account two factors to justify a model working effectively [19]. These factors are one by examining the model's numerical output and two by contrasting its performance with that of other models applied to the same dataset by other studies. As a result, we contrasted the new model's performance with some of the machine learning models employed in earlier studies [20]. We compare the proposed model against KNN, SVM, and Random Forest classifiers. The comparison result is shown in Table 4 below.

Table 4. Comparison of the proposed model with SVM, KNN, Random Forest

Models	Accuracy
Random Forest	72%
KNN	68%
SVM	76%
Proposed model	80%

All the above results shown in Table 4 above are produced with the same dataset and with the same word embedding model, which is FastText. In addition to this, we compared the performance of the proposed idiomatic recognition model (CNN with FastText embedding) with other word embedding models like Term Frequency-Inverse Document Frequency (TF-IDF) and one-hot encoding vectors. The result is depicted as shown in Table 5 below.

Table 5
Comparison of different words vector representation

Model	Word Embedding	Recognition accuracy
CNN	FastText	80%
	TF-IDF	74%
	One-hot encoding	71.3%

According to the results in Table 5 above, CNN with FastText is more effective at identifying idioms in Amharic. This is because the features of idiomatic expressions in the Amharic language can be gained better with the help of FastText's embedding [21].

4. Conclusion

Different NLP models are now being developed for the Amharic language without taking idiomatic expressions into account. This misleads models since idioms' meanings differ from the meanings that may be inferred by looking at words that make them up. Idioms are one of the most fascinating and difficult aspects of Amharic vocabulary. Machine learning algorithms do not process text as input, so they require encoding of texts into another format. We produced a vector of each word used in this study using a pre-trained FastText word embedding as part of this encoding. The experimental findings show that compared to models utilized in this study, the proposed CNN with the FastText embedding model is more effective at detecting Amharic idioms. The proposed approach can therefore be applied to natural language processing tasks requiring the detection of idiomatic expressions, such as machine translation, sentiment analysis, and question-answering systems. However, due to the magnitude of the data, the model's performance requires improvement. Additional information from the holy books of Amharic can be added to the proposed model to enhance its performance. In the future, we plan to conduct Amharic machine translation by incorporating this model as a component of it.

Declarations

Conflict of Interest

The authors declare that they have no conflicts of interest.

Funding

The Authors do not receive funding for this paper.

Data availability

The data can be given upon request from the corresponding author (email: demeke.endalie@ju.edu.et).

Acknowledgment

The authors would like to thank the Jimma institute of technology for supporting them through different resources. The authors would like to thank Jimma University for its support during the research work.

References

1. Debra A Titone, Kyle Lovseth, Kristina Kasparian,Mehrgol Tiv, "Are figurative interpretations of idioms directly retrieved, compositionally built, or both? Evidence from eye movement measures of reading," Canadian Journal of Experimental Psychology, vol. 73, no. 4, p. 216–230, 2019.
2. Oktay Yağiz, "anguage, Culture, Idioms, and Their Relationship with the Foreign Language," Journal of Language Teaching and Research, vol. 4, no. 4, pp. 953-957, 2013.
3. Amsalu Dagnachew, Akililu Worku, □□□□□ □□□□ Idiomatic expressions in Amharic, Addis Ababa, Ethiopia: Kuraz Publishing Agency, 1993.
4. Giancarlo Salton, John D. Kelleher, Robert Ross, "Idiom Token Classification using Sentential Distributed Semantics," in Conference: Annual meeting of the Association for Computational Linguistics (ACL), Berlin, Germany, 2016.
5. Jing Peng , Anna Feldman, "Automatic Idiom Recognition with Word Embeddings," Information Management and Big Data, vol. 656, p. 17–29, 2016.
6. Changsheng Liu, Rebecca Hwa, "A Generalized Idiom Usage Recognition Model Based on Semantic Compatibility," in The Thirty-Third AAAI Conference on Artificial Intelligence, Hawaii, USA., 2019.
7. Anna Feldman, Jing Peng , "Automatic Detection of Idiomatic Clauses," Computational Linguistics and Intelligent Text Processing , vol. 7816, p. 435–446, 2013.
8. Rana Abid Thyab, "The Necessity of idiomatic expressions to English Language learners," International Journal of English and Litrture , vol. 7, no. 7, pp. 106-111, 2016.
9. Haddis Alemayehu, □□□ □□□ □□□□(Love up to the grave), Addis Ababa, Ethiopia: Mega Publishing Agency, 2004.
10. Demeke Endalie, Getamesay Haile, Wondmagegn Taye Abebe, "Feature selection by integrating document frequency with genetic algorithm for Amharic news document classification," PeerJ Computer Science, vol. 8, p. e961, 2022.
11. Martha Yifiru Tachbelie, Wolfgang Menzel, "Amharic Part-of-Speech Tagger for Factored Language Modeling," in International Conference RANLP , Borovets, Bulgaria, 2009 .
12. Michael Gasser, "ornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya," in Conference on Human Language Technology for Development, Alexandria, Egypt, 2011.
13. Haitao Wang,Jie He,Xiaohong Zhang,Shufen Liu, "A Short Text Classification Method Based on N - Gram and CNN," Chinese Journal of Electronics, vol. 29, no. 2, pp. 248-254, 2020.
14. Pedro J. García-Laencina, José-Luis Sancho-Gómez, Aníbal R. Figueiras-Vidal, Michel Verleysen, "K nearest neighbours with mutual information for simultaneous classification and missing data imputation," Neurocomputing, vol. 72, no. 7–9, pp. 1483-1493, 2009.

15. Soudamini Hota, Sudhir Pathak, "KNN classifier based approach for multi-class sentiment analysis of twitter data," International Journal of Engineering & Technology, vol. 7, no. 3, pp. 1372-1375, 2018.
16. Zhi Hong Kok, Abdul Rashid Mohamed Shariff, Meftah Salem M. Alfatni, Siti Khairunniza-Bejo, "Support Vector Machine in Precision Agriculture: A review," Computers and Electronics in Agriculture, Vols. 191,, 2021.
17. Vrushali Y Kulkarni, Pradeep K Sinha, "Random Forest Classifiers :A Survey and Future Research Directions," International Journal of Advanced Computing, vol. 36, no. 1, pp. 1144-1153, 2013.
18. Vaibhavi N Patodkar and Sheikh I.R, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining," International Journal of Advanced Research in Computer and Communication Engineering, vol. 5, pp. 320-322, 2016.
19. Ton Van der Valk, Jan H. Van Driel, Wobbe De Vos, "Common Characteristics of Models in Present-day Scientific Practice," Research in Science Education, vol. 37, no. 4, pp. 469-488, 2020.
20. Pooja Saigal, Vaibhav Khanna , "Multi-category news classification using Support Vector Machine based classifiers," SN Applied Sciences, vol. 2, no. 3, pp. 458-468, 2020.
21. Ben Athiwaratkun, Andrew Wilson, Anima Anandkumar, "Probabilistic FastText for Multi-Sense Word Embeddings," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 2018.

Figures

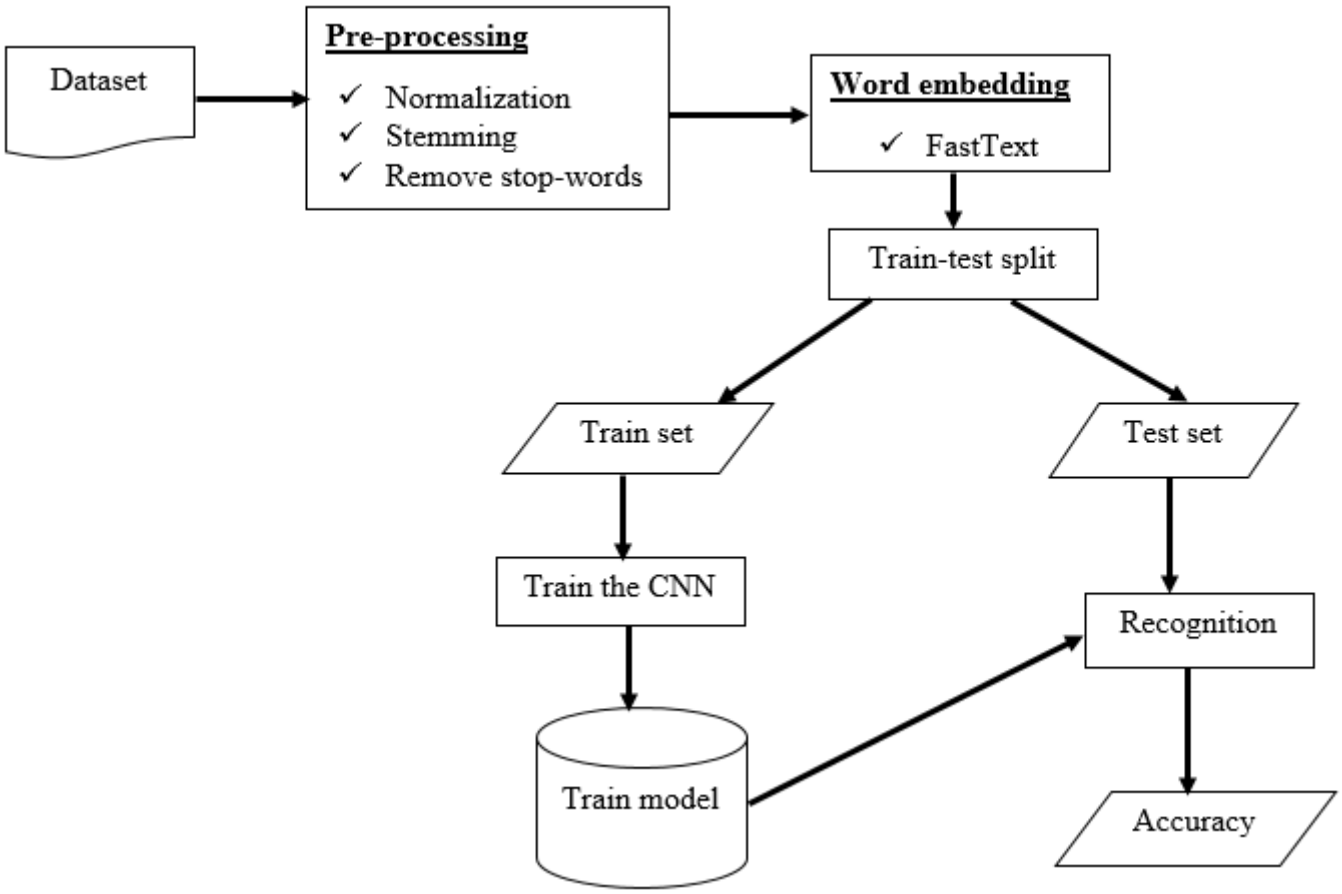

Figure 1

The architecture of the proposed idiomatic expression recognition system

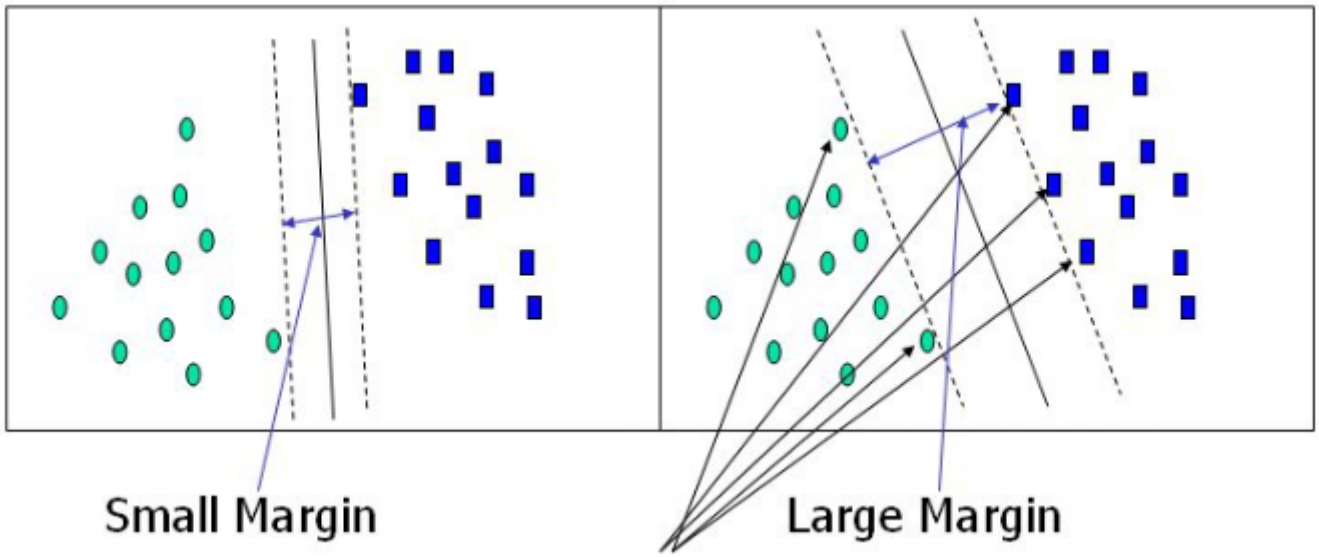

Figure 2

Support Vector

Figure 3

Training Accuracy of the proposed model

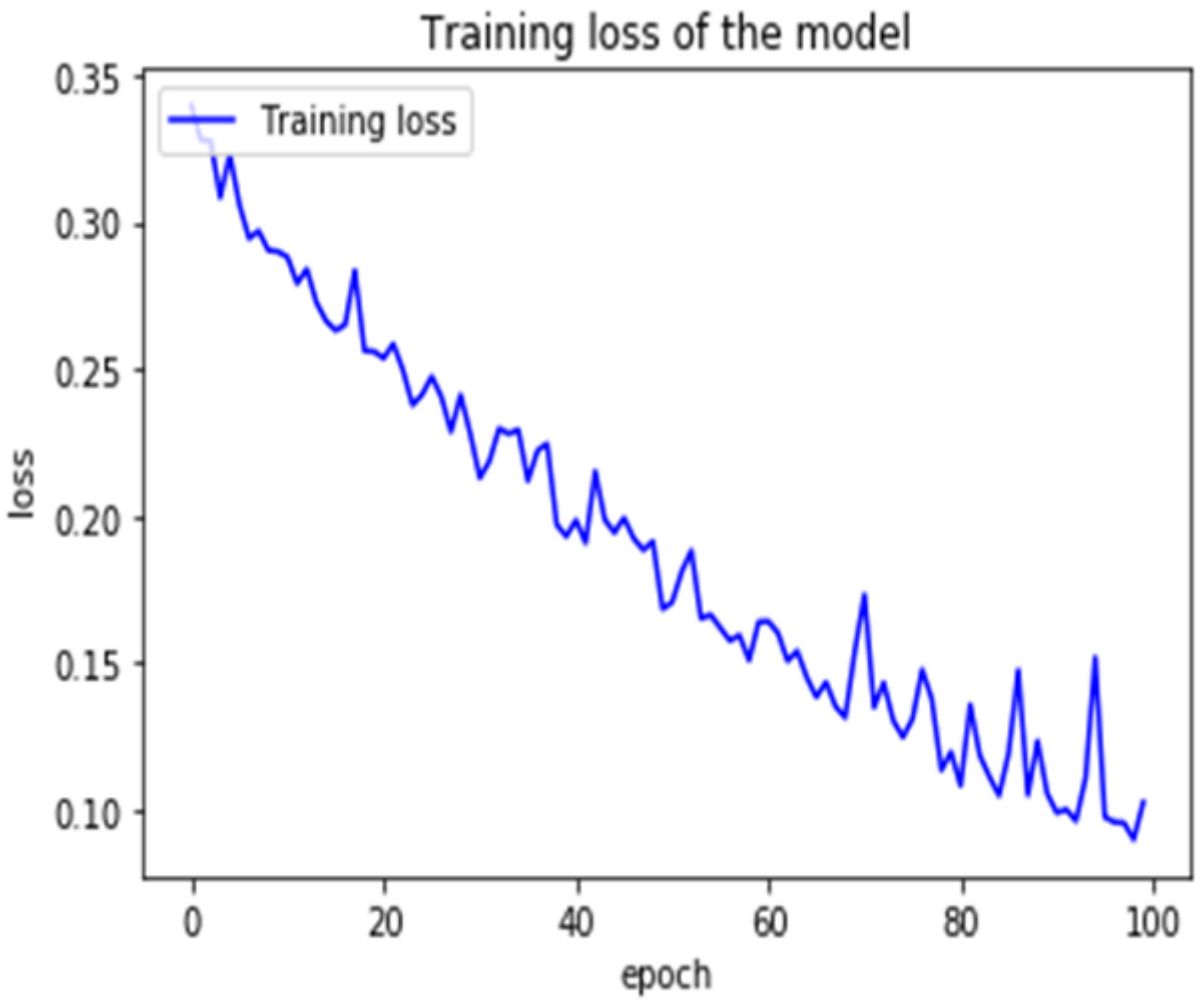

Figure 4

Training loss of the proposed model

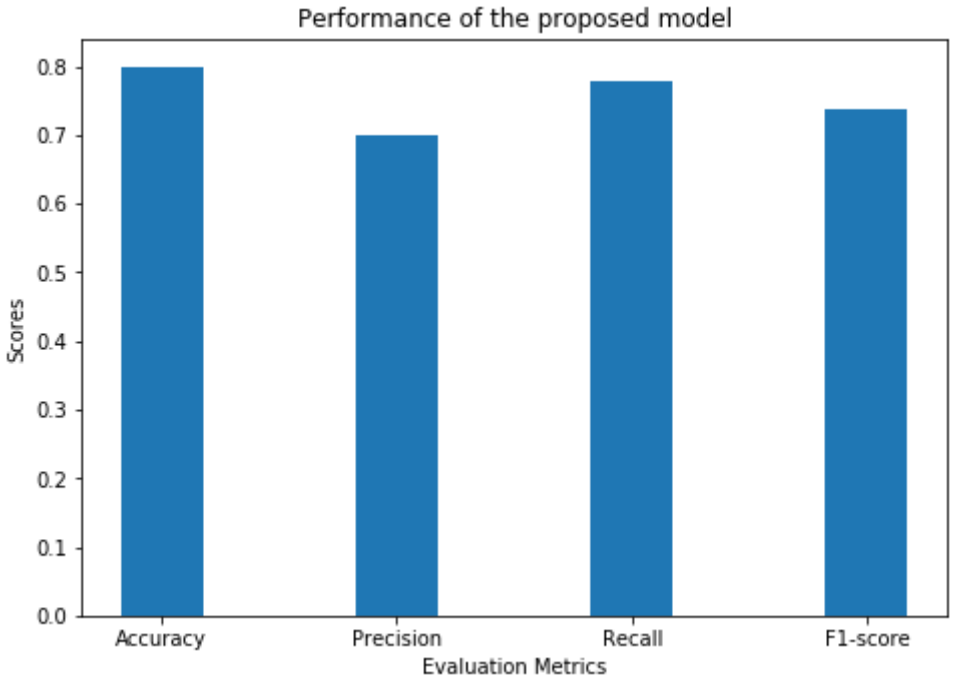

Figure 5

Evaluation of the performance of the proposed model with accuracy, precision, recall and f1-score