Multilingual Recognition Based on Neural Network with High Recognition Degree

Yanhong Jiang¹, *, Xuegang Zhan¹, b, Yixiao Lan², c

¹University of Science and Technology Liaoning School of Computer Science and Software Engineering Anshan, China
²Northeastern University College of Computer Science and Engineering Shenyang, China

*jiangyanhong@ustl.edu.cn, bzhanxuegang@ustl.edu.cn, yixiaolan@foxmail.com

Abstract. Language recognition has become an important branch of natural language processing. Due to the interference of out of vocabulary words and the same words in different languages, it has been a challenging task to accurately identify language categories with high similarity. In this paper, N-gram is combined with word2vec to process the data, MLP and CNN models are selected for improvement and experimental analysis. The experimental results show that the data processing and model training of 20 kinds of highly acquainted languages are used in the above-mentioned way, which reduces the dependence of the model on the size of the data set and the length of the text, the accuracy of the model detection between the languages with high recognition degree is improved, and the robustness of the language recognition model on different data sets is enhanced, with the accuracy rate being over 97.

1. Introduction
Language recognition is to identify which language a given text data belongs to and to detect the language of the data, so that we can recognize the data, understand the semantics of the data, and ultimately obtain the value of the data.

At present, there are many open source tools for language recognition, Such as Liu et al. Created langid.py in 2012 using n-gram features combined with polynomial naive Bayesian classifiers [1]; Nakatani released IDIG in the same year, a language recognition tool based on common sense, regularization and Bayesian classifier [2], the tool is used for Twitter datasets; In 2013, Brown proposed a spatial vector model based on N-gram character feature weights [3]; There is also the commonly used open source language-detection tool, which builds its own dictionary through loaded language packs, accuracy is related to the number of language samples loaded by the user, the more samples of each language, the higher the accuracy of detection.

The traditional language recognition based on statistics is highly dependent on the dataset, and each language also has word/word deformation and synthetic abbreviations, which is also difficulty in language detection. There are also some common words between different languages, such as Japanese has many Chinese characters, French also has many English words, which will affect the accuracy of language detection. Baldwin et al. pointed out that achieving good recognition on six European language collections does not mean that they will achieve the same good results in a collection of corpora with...
more languages [4]. Liu et al. Pointed out that the accuracy of the same model varies greatly on different datasets [5]. They also point out that removing noise from the dataset can significantly improve the recognition rate.

As we all know, the more kinds of languages, the more difficult the language detection is, and the lower accuracy rate is. Although language recognition has been widely studied, most of them focus on individual languages. Such as Cav-nar and Trenkle approached the task of language classification in monolingual documents in [6], by using N-gram analysis. This paper studies the recognition of 20 languages with high similarity. In order to verify that the neural network has high accuracy among multilingual recognition with high similarity, this paper uses n-gram combined with word2vec to process the data and chooses MLP and CNN models for improvement and experimental analysis.

2. Related WORK

As far as being aware and up to our knowledge, no work has been published for using MLP, CNN models on Multilingual language recognition with high similarity and we propose to use N-gram model in conjunction with word2vec to process data. The methods and models used in this article are described below.

2.1. N-gram

N-gram (also called N-gram model) is a very common and important concept in natural language processing. N-gram assumes that the nth word appears to be related to the first n-1 word and not to any other word. The probability of the whole sentence is equal to the product of the probability of each word. The probability of each word can be calculated by statistics in the corpus. Assuming that the sentence S is composed of the word sequences W_1, W_2, W_3 … W_n, the N-gram language model is formulated as follows:

\[ P(S) = P(W_1, W_2, \ldots, W_n) = P(W_1)P(W_2 \mid W_1)\ldots P(W_n \mid W_{n-1}) \]  

The commonly used N-Gram models are Bi-Gram and Trigram This paper uses Bi-gram method to divide a complete sentence into several grams. In the course of the experiment, we take N=2 (Bi gram) For example, given the following sentence: Summer is hot., will be divided into: Su um m mm me er…

2.2. word2vec

In 2013, Google opened up a tool for word vector computing: word2vec [7],[8]. One of the many features of the tool is that word2vec can be trained efficiently on millions of dictionaries and hundreds of millions of datasets; Secondly, word embedding, the training result of the tool, can measure the similarity between quantifiers. Two important models are mentioned in word2vec: CBOW and Skip-Gram. About the two models Author Tomas Mikolon gives schematic diagrams as shown in Fig. 1 and Fig. 2 in [9]. As can be seen from the figure, both models have three layers: the input layer, the projection layer, and the output layer. The former predicts the current word W_i on the premise that the context W_{i-2}, W_{i-1}, W_{i+1}, W_{i+2} of the current word W_i is known(See Fig. 1);The latter is exactly the opposite, which is to predict the context W_{i-2}, W_{i-1}, W_{i+1}, W_{i+2} of the current word W_i is known(See Fig. 2).
2.3. **Multilayer perceptron**

Multilayer Perceptron (MLP) is a generalization of perceptron, which overcomes the weakness of perceptron in recognizing linear non-separable data [10]. For MLP, we refer to the first layer as the input layer, the last layer to the output layer, and the middle layer as the hidden layer. The MLP does not specify the number of hidden layers, so the appropriate number of hidden layers can be selected according to their respective needs, and there is no limit to the number of neurons in the output layer. In the Fig. 3, only one hidden layer is involved, input n variables \([x_1, x_2, x_3, \ldots x_n]\), output layer has n neurons. In this paper, Relu is selected as the activation function, which can effectively overcome the problem of gradient disappearance and speed up the training [11].
2.4. Convolutional Neural Network
Convolutional Neural Networks (CNN) is a class of feedforward neural networks with convolutional computation and deep structure. It is one of the representative algorithms for deep learning [12], [13]. A convolution neural network consists of several convolution layers, Pooling layers and full connection layers. Convolutional layer: Each convolutional layer in a convolutional neural network consists of several convolution units, and the parameters of each convolution unit are optimized by back propagation algorithm. The purpose of convolution operation is to extract different features of input. Pooling layer: It is actually a form of desampling. Usually, after the convolutional layer, a feature with a large dimension is obtained. Cut the feature into several areas, take its maximum or average value to get new, smaller dimension of the feature. Fully connected layer: Combines all local features into global features to calculate the score for each final class.

3. Data preparation
3.1. Data Sources
This article trains MLP, CNN models in 20 highly acquainted languages. The datasets for English, Filipino, Indonesian, and Malay are from the Asian Language Treebank (ALT). The ALT project aims to promote the most advanced Asian Natural Language Processing (NLP) technology through the development and use of ALT’s open collaboration [14]. Data sets in Czech, Danish, German, Spanish, Estonian, Finnish, French, Italian, Lithuanian, Latvian, Dutch, Polish, Portuguese, Slovak, Slovenian, Swedish and Hungarian are from the European Parliament Proceedings Parallel Corpus [15]. The parallel corpus of the European Parliament, drawn from the minutes of meetings of the European Parliament and covering the period from 1996 to 2006, is still being expanded. The abbreviations for each language are shown in table 1.

| Language       | Abbreviation |
|---------------|--------------|
| English: en   | en           |
| Hungarian: hu | hu           |
| Indonesian: id| id           |
| Malay: ms     | ms           |
| Czech: cs     | cs           |
| Danish: da    | da           |
| German: de    | de           |
| Spanish: es   | es           |
| Finnish: fi   | fi           |
| French: fr    | fr           |
| Italian: it   | it           |
| Lithuanian: lt| lt           |
| Dutch: nl     | nl           |
| Polish: pl    | pl           |
| Estonian: et  | et           |
| Latvian: lv   | lv           |
| Portuguese: pt.| pt.         |
| Slovak: sk    | sk           |
| Slovenian: sl | sl           |
| Swedish: sv   | sv           |
3.2. Data preprocessing

Firstly, the data in each language is cleaned, the extra characters and identifiers are removed. Such as:

- cs data: "... bere na vědomí alarmující pokles pomoci EU z 47,7 miliard EUR v roce 2006 na 46,1 miliard v roce 2007 .....;"
- ms data: SNT.193201.29 Didalam kedudukan mata, Harvick dan Johnson kekal ditempat pertama dan kedua.

Other characters and identifiers in the data, such as ... and SNT.193201. 29, are meaningless for data analysis and therefore need to be cleaned out.

Secondly, we have also reserved for data that is too short in length. Short data in other languages such as: fi data: Keskiviikon osalta; pl data: Otwarcie posiedzenia. The purpose was to verify the accuracy of the model identification used in this experiment.

Finally, Corresponding training data amounts are selected for 20 languages respectively: 1000, 5000, 10000, 50000, 100000, 200000 and different languages with the same data amount are stored randomly in Language.txt. Another file, Target.txt, holds the corresponding language kind. The data of 200000 is selected from the corpus as the test corpus, and the storage method is consistent with the training corpus. As shown in Table 2.

| Language.txt | Target.txt |
|--------------|------------|
| Italia berhasil mengalahkan Portugal 31-5 di grup C dalam Piala Dunia Rugby 2007 di Parc des Princes, Paris, Perancis. | Id |
| A Parlament állásfoglalásaival kapcsolatos további intézkedések: lásd a jegyzőkönyvet | Hu |
| Veiksmai įgyvendinant Parlamento rezoliucijas | Lt |
| Ripresa della sessione | It |
| ... | ... |

4. Experiment

4.1. gram2vec

We use N-gram method to divide a complete sentence into several tokens. In the course of the experiment, we take n = 2 (Bi-gram). Because the tokens being modeled to express a sentence are not words but grams, we call it gram2vec.

Then, we use word2vec’s CBOW model to get the gram vector for each sentence, five grams on each side of a particular gram are selected from each gram sequence to predict the vector of the central gram, and outcomes each gram with its own vector. We set the CBOW model to eventually generate 300-dimensional vector, 500-dimensional vector and 700-dimensional vector for each gram and set the number of iterations to 5. The first 200000 rows of data in each language are modeled as gram2vec model datasets, and each language dataset is modeled as a union.

4.2. Experiments with MLP

When building the MLP model, set the model as an input layer, two hidden layers, and an output layer. Where the input dimensions are 300, 500 and 700 respectively, the activation function is Relu, and 50% of the neurons are randomly disconnected when the next layer is connected to prevent over-fitting; The number of neurons in the hidden layer is 128. In the output layer, the number of neurons is the number of languages, using Softmax as the activation function. The model uses stochastic gradient descent algorithm for backpropagation [16],[17]. Because this paper studies the classification of 20 languages with high recognition, categorical_crossentrophy is chosen as the loss function [18].

The input of the model is the sentence vector and the corresponding label of the sentence vector. The zero vectors of 300, 500 and 700 dimensions are established as the initial sentence vectors. Then, the model finds out whether each gram in the sentence exists in the pre-trained gram2vec model. If it exists, add the gram corresponding vector to the sentence vector to get a new sentence vector. Finally, count
the number V of Bi-gram in the gram2vec vocabulary table and divide the sentence vector by V to get the final sentence vector. The label corresponding to each sentence vector is a one-hot vector, and the size of the dimension is the number of languages. The obtained sentence vector and the corresponding label are sent to the MLP for training, and the batch size is set to 128.

The experimental results are shown in the Table 3.

Table 3. Experimental results of MLP

| Amount of data*20 | Word vector dimension | Method      | Accuracy rate |
|-------------------|-----------------------|-------------|---------------|
| 1000              | 300                   | W2V+MLP     | 0.895         |
| 5000              | 300                   | W2V+MLP     | 0.9304        |
| 10000             | 300                   | W2V+MLP     | 0.9146        |
| 50000             | 300                   | W2V+MLP     | 0.9722        |
| 100000            | 300                   | W2V+MLP     | 0.9773        |
| 200000            | 300                   | W2V+MLP     | 0.9706        |

| Amount of data*20 | Word vector dimension | Method      | Accuracy rate |
|-------------------|-----------------------|-------------|---------------|
| 1000              | 500                   | W2V+MLP     | 0.909         |
| 5000              | 500                   | W2V+MLP     | 0.939         |
| 10000             | 500                   | W2V+MLP     | 0.9147        |
| 50000             | 500                   | W2V+MLP     | 0.9738        |
| 100000            | 500                   | W2V+MLP     | 0.9773        |
| 200000            | 500                   | W2V+MLP     | 0.9726        |

| Amount of data*20 | Word vector dimension | Method      | Accuracy rate |
|-------------------|-----------------------|-------------|---------------|
| 1000              | 700                   | W2V+MLP     | 0.93          |
| 5000              | 700                   | W2V+MLP     | 0.9362        |
| 10000             | 700                   | W2V+MLP     | 0.935         |
| 50000             | 700                   | W2V+MLP     | 0.97715       |
| 100000            | 700                   | W2V+MLP     | 0.97308       |
| 200000            | 700                   | W2V+MLP     | 0.9738        |

MLP is a multi-layer perceptron, and every element of the word vector is connected to the neural network. In this case, the fitting ability of the whole model is relatively insufficient, sensitive to the random characteristics of the data, and dependent on the uniform characteristics of different types of data. The convergence times of training are more, and the risk of over-fitting is greater. From the experimental data, it can be concluded that under the same amount of data, the larger the dimension of the word vector, the higher the correct rate. When the data amount is 200000*20 and the word vector dimension is 700, the accuracy rate is the highest. Under the same word vector dimension, the larger the amount of data, the better the effect (as shown in Fig. 4). During the experiment, the correct rate has dropped. The reason may be that the amount of data is too large to fit. However, if there are more training rounds, there is a risk of fitting. It can be seen that when the vector dimension reaches 700, a relatively good performance can be achieved with a small amount of data.
4.3. Experiments with CNN

The CNN neural network model consists of four one-dimensional convolution neural networks, a maximum pool layer, a global average pool layer, a Dropout layer and a fully connected neural network. The first layer is a one-dimensional convolution neural network with 64 filters. The convolution kernel size is 3 and the activation function is Relu. The second layer is the same as the first layer; The third layer is a maximum pooling layer with a convolution core size of 3; The fourth layer is a one-dimensional convolution neural network with 128 filters. The convolution kernel size is 3 and Relu is used as the activation function. The fifth is the same as the fourth; The sixth layer is a global average pooling layer; Layer 7 is a Dropout layer that randomly disconnects 50% of the neurons from the next layer; The last layer is the output layer, a full connection layer with 20 neurons, and the activation function used is Sigmoid.

The model uses categorical crossentry as a loss function and RMSprop for backpropagation. The way to turn a sentence into a sentence vector is to divide the sentence into Bi-gram first. Take out the first 30 grams in order to form a sequence. The gram in the sequence is replaced by the corresponding vector of the gram in the trained gram2vec model, that is, the vector representing the sentence is obtained (see Fig. 5). If the Bi-gram corresponding to the sentence is less than 30 grams, the zero vector of the same dimension as the gram corresponding vector will be used to complement it. 128 samples are sent in each training epoch. The training label is a 20 dimensional one-hot vector, with each dimension representing a language.
I like Beijing ...

Gram2Vec Model
Gram2Vec Model
Gram2Vec Model
Gram2Vec Model

GramVector
GramVector
GramVector
GramVector

Sentence vector

**Figure 5.** Sentence vector representation process

The experimental results are shown in the Table 4.

**Table 4.** Experimental results of CNN

| Amount of data*20 | Word vector dimension | Method   | Accuracy rate |
|-------------------|-----------------------|----------|---------------|
| 1000              | 300                   | W2V+CNN  | 0.973         |
| 5000              | 300                   | W2V+CNN  | 0.9498        |
| 10000             | 300                   | W2V+CNN  | 0.9725        |
| 50000             | 300                   | W2V+CNN  | 0.9863        |
| 100000            | 300                   | W2V+CNN  | 0.95          |
| 200000            | 300                   | W2V+CNN  | 0.9862        |

| Amount of data*20 | Word vector dimension | Method   | Accuracy rate |
|-------------------|-----------------------|----------|---------------|
| 1000              | 500                   | W2V+CNN  | 0.9749        |
| 5000              | 500                   | W2V+CNN  | 0.973         |
| 10000             | 500                   | W2V+CNN  | 0.9724        |
| 50000             | 500                   | W2V+CNN  | 0.9561        |
| 100000            | 500                   | W2V+CNN  | 0.9498        |
| 200000            | 500                   | W2V+CNN  | 0.95          |

| Amount of data*20 | Word vector dimension | Method   | Accuracy rate |
|-------------------|-----------------------|----------|---------------|
| 1000              | 700                   | W2V+CNN  | 0.9711        |
| 5000              | 700                   | W2V+CNN  | 0.9854        |
| 10000             | 700                   | W2V+CNN  | 0.9499        |
| 50000             | 700                   | W2V+CNN  | 0.9864        |
| 100000            | 700                   | W2V+CNN  | 0.9854        |
| 200000            | 700                   | W2V+CNN  | 0.9861        |

The convolutional neural network is characterized by sparse connections and multi-feature maps, using multi-layer convolutional neural networks to link word vectors to find the characteristics of vectors.
representing different languages, and then to classify them. Similar to the classification of images, the feature matrix of a language does not need to require that the data of each category have very consistent characteristics but has certain characteristics. Therefore, this feature of convolutional neural networks is well suited for use in language classification. From the experimental results: When the amount of data is 200000*20, the relationship between dimensions and accuracy is not positively correlated. However, convolutional neural networks can achieve good accuracy even in low-resource situations. In the same dimension, the accuracy rate fluctuations for different data volumes are relatively random (as shown in Fig. 6). On the one hand, the reason is that the characteristic matrix of a certain language is changed because of the different amount of data, but for some characteristic changes, convolutional neural network does not capture them in time; On the other hand, with the increase of the amount of data, the underfitting phenomenon appears, which will be reduced by adjusting the hyperparameters properly.

![Graph](image1.jpg)

Figure 6. Comparison results of different dimensions of CNN

5. Conclusion
In this paper, word2vec is combined with MLP and CNN models to identify multi-lingual recognition with high degree of recognition, which improves the recognition performance of the model. The accuracy of recognition between languages with high degree of recognition is over 97%. Compared with the traditional language recognition method and the existing language recognition tools Langid and Langdetect, the model not only reduces the dependence on the length of data and the size of data sets, but also has good recognition effect in practical application.

References
[1] Lui M. Baldwin T. langid.py: An Off-the-shelf Language Identification Tool. In: Proc. of ACL 2012 System Demonstrations. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012. 25 - 30.
[2] Nakagami Hideo. Short Text Language Detection with Infinity-Gram. Nara Institute of Science and Technology, 2012.
[3] Brown R D. Selecting and weighting n-grams to identify 1100 language. Speech, and Dialogue. Lecture Notes in Computer Science. Berlin, Heidelberg, Germany: Springer,2013.475-483.
[4] Baldwin T. Liu M. Language identification: The long and the short of matter. In: Human
Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010. 229-237.

[5] Liu M, Baldwin T. Accurate language identification of twitter messages. In: Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM). Gothenburg, Sweden: Association for Computational Linguistics, 2014. 17-25.

[6] Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: Proceedings of SDAIR 1994, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, US, pp.161-175 (1994).

[7] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013, January 17). Efficient Estimation of Word Representations in Vector Space. arXiv.org.

[8] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013, October 17). Distributed Representations of Words and Phrases and their Compositionality. arXiv.org.

[9] Tomas Mikolov, Quoc V. Le, Ilya Sutskever. Exploiting Similarities among Languages for Machine Translation. arXiv:1309.4168v1, 2013.

[10] Neural Networks for pattern recognition. Oxford University Press. 1995. ISBN 0198538642.

[11] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems. 2012, 1: 1097–1105.

[12] Goodfellow, I., Bengio, Y., Courville, A. Deep learning (Vol. 1). Cambridge: MIT press, 2016: 326-366.

[13] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahrourdy, A., Shuai, B., Liu, T., Wang, X., Wang, L., Wang, G. and Cai, J., 2015. Recent advances in convolutional neural networks. arXiv preprint arXiv:1512.07108.

[14] Riza H, Purwoadi M, Gunarso, et al. Introduction of the Asian Language Treebank[C]// Coordination & Standardization of Speech Databases & Assessment Techniques. IEEE, 2017.

[15] Europarl: A Parallel Corpus for Statistical Machine Translation, Philipp Koehn, MT Summit 2005.

[16] Rosenblatt, Frank. x. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961.

[17] Rumelhart, David E., Geoffrey E. Hinton, and R. J. Williams. Learning Internal Representations by Error Propagation. David E. Rumelhart, James L. McClelland, and the PDP research group. (editors), Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundations. MIT Press, 1986.

[18] Michael Jordan, Jon Kleinberg, Bernhard Schölkopf. Pattern Recognition and Machine Learning; Springer Science+Business Media, LLC; Singapore, 2006; pp.244-267.