A Comparative Study of N-Grams, Word2Vec and GloVe Embeddings For Language Identification

Natural Language Processing, *Bruno Esteves, UnB,*

***Abstract*—Language Identification is a task in Natural Language Processing that involves identifying the native language of a given text or conversation. This paper describes an experiment using Convolutional Neural Networks alongside Language Models to train a classification model that successfully predicts a document language based on a 22 language *corpora* with 22000 documents. Experiments were conducted to assess not only the efficiency of n-grams model languages but also to observe the Word2Vec and GloVe's behaviour in this area of study.**

***Index Terms*—LID, NLP, N-Grams, Word2Vec, GloVe, CNN**

I. Introduction

LANGUAGE identification (LID) is a task in the area of Natural Language Processing (NLP) that involves identifying the native language of a given text or conversation. With the accelerated globalization of the last decades, LID has become a necessity for communication between different languages. News from the other side of the world travels fast in various forms of text, image, audio and video. All of these require a different kind of approach in the LID field. One of the main problems of natural language processing is automatic text translation, and language identification is the basis for this type of task.

Within the area of study of natural language processing we have text categorization, or classification. Sebastiani et al. [1] describes text categorization, which can be simplified to mapping a text to one or more values from a predefined set of labels or classes. Past research has been focused on monolingual documents [2], where a text has only one native language and it is categorized to only one class. Therefore we limit the scope of our discussion to monolingual documents.

In the history of LID, many classifiers were built based on statistical inferences of word frequency in documents. The use of n-grams was widely present in works that studied Co-ocurrences analysis to calculate co-occurrence ratios of any n-characters, and n-grams repetition to observe its frequency among documents.

To delve deeper into the classification problem, we need to discuss neural networks. Neural networks are a set of algorithms modeled based on the human brain that are designed to recognize patterns. Wang et al. [3] describes that a neural network consists of an input layer of neurons (or nodes, units), a set of hidden layers of neurons, and a final layer of output neurons. The kind of neural network we are interested in this study is the Convolutional Neural Network (CNN). They are useful in a lot of applications, especially in image related tasks, but in recent years CNN has demonstrated to be successful in text classification tasks. The N-gram language model has led to many outstanding performances on deep neural networks. Moreover, some word embedding algorithms like GloVe and Word2Vec are likely to produce a state of performance achieved by neural networks. In this study we make a comparative of those embedding techniques used with a CNN in a LID task.

This paper is structured as follows. Section II presents the related works on the use of n-grams and CNN for text classification. Section III details the proposed method and its constituting steps. Section IV describes the experimental results that were conducted on a multilanguage dataset using qualitative evaluation metrics. Section V concludes this paper and discusses possibilities for future work.

II. Related Works

The LID task can not only try to distinguish different languages among texts, but also different varieties of the same language. Zampieri et al. [4] presented a supervised computational methods for the identification of Spanish language varieties using n-gram language model. Four journalistic corpora from different countries were used in these experiments : Spain, Argentina, Mexico and Peru.

Haitao Wang et al. [5] explored the problem of short text classification using CNN by adopting none linear sliding method and N-gram language model, and picks out the key features by using the concentration mechanism.

Kalchbrenner et al. [6] used a CNN for modelling sentences in four different experiments: small scale binary and multi-class sentiment prediction, six-way question classification and Twitter sentiment prediction by distant supervision.

III. Proposed Method

The proposed method is divided into the following steps: dataset definition, preprocessing, definition of the different language models selected for comparison and evaluation, and definition of the Neural Network model we used in this experiment.

A. Dataset Definition

The choice of dataset was made considering the diversity of languages present, not only more used but also some less common in LID studies. It contains 22 different languages with 1000 exemples each. This dataset is a subset of the WiLi-2018 wikipedia dataset.

TABLE I
DESCRIPTION OF THE DATASET

Languages	Quantity
English	1000
Arabic	1000
French	1000
Hindi	1000
Urdu	1000
Portuguese	1000
Persian	1000
Pushto	1000
Spanish	1000
Korean	1000
Tamil	1000
Turkish	1000
Estonian	1000
Russian	1000
Romanian	1000
Chinese	1000
Swedish	1000
Latin	1000
Indonesian	1000
Dutch	1000
Japanese	1000
Thai	1000

B. Dataset Pre-Processing

Languages like Chinese, Japanese and Thai can have texts without any blank spaces, only a continuous string of characters. These strings can reach over 30 characters, and withou pre-processing embeddings and common tokenizers will not be able to function with it. So a specific Python package was used to remove stop-words and pre-process those 3 languages, alongside with the basic pre-processing for the rest of the *corpora*.

- *pythainlp* for Thai.
- *advertools* for Chinese and Japanese.
- *re* for the rest of pre-processing.

C. N-Grams

Qafmolla et al. [7] described an n-gram as an n-character slice of a longer string containing blanks to the beginning and ending so as to categorize it as a beginning or ending of-word-n-gram. The representation in N-Grams of the word *Translation* can be represented in the uni-gram, bi-gram and tri-gram respectively as it follows:

"*Translation*"

uni-gram : **T, r, a, n, s, l, a, t, i, o, n**

bi-gram : **_T, Tr, ra, an, ns, sl, la, at, ti, io, on, n_**

tri-gram : **__T, _Tr, ran, ans, nsl, sla, lat, ati, tio, ion, on_, n__**

Fig. 1. N-gram representation

D. Word2Vec

Word2Vec is a statistical method for efficiently learning an independent Word Embedding from a text corpus. A Word2Vec language model was trained from our training dataset with the Skip-Gram architecture. In this architecture it tries to find the context words based on a given central word.

E. GloVe

GloVe stands for Global Vectors for word representation. It is an unsupervised learning algorithm that generates word embeddings by aggregating global word co-occurrence matrices from a given corpus. It generates relationships between the words from statistics based on these matrices and each value in the co-occurrence matrix represents a pair of words occurring together.

F. CNN Training

The CNN used in the experiment is architected as it follows:

- *Embedding* layer
- *1D Convolution* layer
- *Dense* layer
- *Dense* layer
- *Output* layer

The network's first layer is the *Embedding* layer. To input the layer's values we first need to specify the size of the considered vocabulary. The network is only aware of the training data. Validation and test data are unknown to the model. For each experiment the size of the vocabulary varies due to its different architectures.

For the input size of the network we decided a fixed length of 64 based on the statistical analysis of the quantity of words in each corpus sentence. In cases when the input is shorter than the defined length we use a technique called *padding*. The missing words are filled with a default character called *OOV* (Out of Vocabulary).

TABLE II
VOCABULARY SIZE COMPARISON

Model	Vocabulary Size
3-Gram	352124
5-Gram	1361798
Word2Vec	228705
GloVe	229097

Afterwards there is the Convolution layer, that takes the output of the *Embedding* layer and creates a convolution kernel that is convolved with the layer input over a single spatial dimension to produce the outputs. For each of the experiments to follow the same pattern, we fixated the kernel size to 6 and strides to 1.

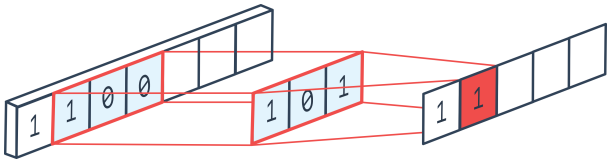

Fig. 2. Example: 1D Convolutional Layer with kernel = 3 and strides = 1

Next in sequence we have two Dense layers. They are simply a densely-connected neural network layer and have

the responsibility to learn and train its weights so its output is propagated to the output layer. Both of them are activated by a *ReLU* function. The rectified linear activation function is a linear function that will output the input directly if it is positive, otherwise, it will output zero.

Finally we have the output layer, which is also a Dense layer but its activation is defined by another type of function called *Softmax*. This function transforms the output into values between 0 and 1, so that they can be interpreted as probabilities. The output size of the last layer is equal to the number of classes existing in our classification problem.

IV. Experiments Results

Here we evaluate the experiments results to make the comparison between them. Each of the models were trained using standardized parameters:

- Learning Rate : 0.02
- 3 Dropout Layers : 0.5/0.3/0.3
- Epochs : 50

Since the learning rate is considered a high value we are using Dropout Layers for each of the Activation Layers to prevent overfitting the model. To evaluate the performance of each model we use the *Precision*, *Recall* and *F1-Score*.

A. Precision

Precision is a measure of how many of the class predictions made are correct.

$$\frac{N.ofCorrectlyPredictedClassInstances}{N.ofTotalClassPredictionsByTheModel} \quad (1)$$

B. Recall

Recall is a measure of how many of the class cases the classifier correctly predicted, over all the positive cases in the dataset.

$$\frac{N.ofCorrectlyPredictedClassInstances}{N.ofTotalClassInstancesInTheDataset} \quad (2)$$

C. F1-Score

F1-Score is a measure combining both precision and recall, or the harmonic mean of the two.

$$2 * \frac{Precision * Recall}{Precision + Recall} \quad (3)$$

After all the experiments we aggregated the results side by side evaluate their performance. All results were satisfactory given their metrics, with the 3-gram model being the one with the highest value. The Word2Vec and GloVe embeddings both proved to be efficient in the Language Identification tasks with good metrics. The 5-gram model evaluation was below the expectations, leading to the idea that the more we increase the "slice" of the n-gram the lesser is the efficiency.

Analyzing the loss of the models in each experiment, we can observe that the models based on n-gram have a longer interval between the beginning of training and the fall of the Loss curve, which implies a delay for the learning weights to start to adjust efficiently.

TABLE III
Precision, Recall and F1 Score

Model	Precision	Recall	F1 Score
3-Gram	0.95	0.94	0.94
5-Gram	0.86	0.83	0.84
Word2Vec	0.92	0.92	0.91
GloVe	0.94	0.91	0.90

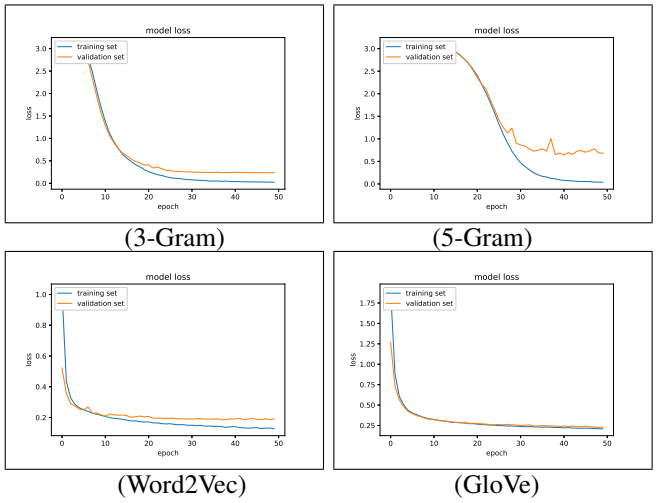

Fig. 3. Loss history in the models training.

Comparing the Confusion Matrix from the 3-gram model to the others we can observe that it was the most successful in concentrating the values in its main diagonal, meaning that if was the most successful in correct predictions. The other models have in common the concentration of wrong predictions in the second row and column. The class with index 2 represents the Chinese language and it was the one with most wrong predictions. The 5-gram model predicted most of the Chinese instances as it was Thai (index 19), and had a difficulty predicting the Japanese (index 8) class. The Word2Vec and GloVe embeddings also couldn't be successful with the Chinese class predictions. Word2Vec predicted most Chinese instances as Japanese, and GloVe predicted most Japanese instances as Chinese.

V. Conclusion

This study explored the use of different language models and embeddings to evaluate its efficiency in the Language Identification taks. The method using a Convolutional Neural Network with a 1D convolutional layer to work with text data was proposed in order to compare its behaviour when training and evaluate its metrics. The 3-gram model was the best one in general, specially in differentiating and classifying correctly the the Chinese and Japanese languages. This is due to the fact that the 3-gram architecture is able to separate the ideograms in a smaller window making it easier to make the distinction from one ideogram to another. Future work can be guided to use a different kind of Neural Network alongside different types of word embeddings in the LID task to observe its behaviour.

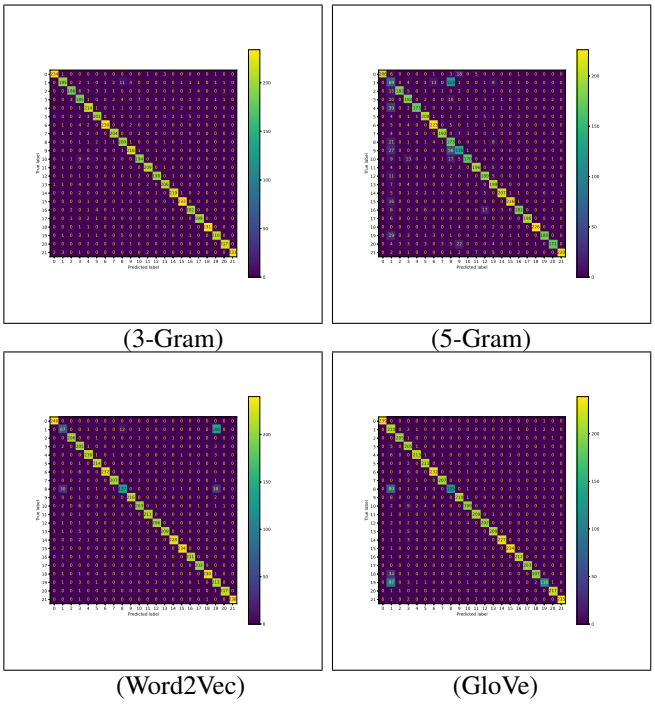

Fig. 4. Confusion Matrices plotted in the experiments.

REFERENCES

[1] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, p. 1–47, mar 2002. [Online]. Available: https://doi.org/10.1145/505282.505283

[2] B. Hughes, T. Baldwin, S. Bird, J. Nicholson, and A. MacKinlay, "Reconsidering language identification for written language resources," in *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. Genoa, Italy: European Language Resources Association (ELRA), May 2006. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2006/pdf/459_pdf.pdf

[3] S.-C. Wang, *Artificial Neural Network*. Boston, MA: Springer US, 2003, pp. 81–100. [Online]. Available: https://doi.org/10.1007/978-1-4615-0377-4_5

[4] M. Zampieri, B. G. Gebre, and S. Diwersy, "N-gram language models and POS distribution for the identification of Spanish varieties (ngrammes et traits morphosyntaxiques pour la identification de variétés de l'espagnol) [in French]," in *Proceedings of TALN 2013 (Volume 2: Short Papers)*. Les Sables d'Olonne, France: ATALA, Jun. 2013, pp. 580–587. [Online]. Available: https://aclanthology.org/F13-2010

[5] H. Wang, J. He, X. Zhang, and S. Liu, "A short text classification method based on n-gram and cnn," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 248–254, 2020. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/cje.2020.01.001

[6] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 655–665. [Online]. Available: https://aclanthology.org/P14-1062

[7] N. Qafmolla, "Automatic language identification," *European Journal of Language and Literature*, vol. 3, no. 1, p. 140–150, Jan. 2017. [Online]. Available: https://revistia.org/index.php/ejls/article/view/5752