Journal Homepage: - www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI: 10.21474/IJAR01/14012
DOI URL: http://dx.doi.org/10.21474/IJAR01/14012

RESEARCH ARTICLE

CLASSIFICATION OF BAOULÉ SENTENCES ACCORDING TO FREQUENCY AND SEGMENTATION OF TERMS VIA CONVOLUTIONAL NEURAL NETWORKS

Hyacinthe Kouassi Konan[1], Francis Adlès Kouassi[1], Guy L. Diety[3] and Olivier Asseu[1,2]

1. Ecole Supérieure Africaine Des Technologies d'Information Et De La Communication (ESATIC), Côte d'Ivoire.
2. Institut National Polytechnique Félix Houphouët Boigny (INP-HB), Côte d'Ivoire.
3. Université Virtuellede Côte d'Ivoire (UVCI).

Manuscript Info

Manuscript History
Received: 05 November 2021
Final Accepted: 09 December 2021
Published: January 2022

Key words:-
Classification of Sentences, CNN, Frequency of Terms, Segmentation

Abstract

In the Baoulé language, several sentences express the same fact. Classification of sentences is a task of Natural Language Processing (NLP). Deep learning has turned out to be a kind of method that has a significant effect in this area. In this paper, we propose a convolutional neural network (CNN) based system for sentence classification. We introduce into this system a word representation model to capture semantic characteristics by encoding the frequency of terms and segmenting the sentence into clauses. The experimental results show that our system produces satisfactory results.

Introduction:-

Thanks to the rapid development of ICTs, the media generate a great deal of information on a daily basis in the African language (for example in the Baoulé language), in particular original articles, news, reviews, etc. This information is disseminated in the form of text and is used to judge the effectiveness of the information. Thus, in recent years, sentence classification has been activated in the field of Natural Language Processing (NLP).

The traditional methods of classification of sentences are mainly based on statistical principles, and which are almost classic machine learning methods.

Currently, deep learning models have obtained remarkable results in computer vision **[1]**, speech recognition **[2]** and NLP **[3]**. CNNs have been shown to be very effective in obtaining satisfactory results for sentence classification tasks.

In a sentence classification task, the objective is to predict the class tag information for one or more sentences. For example, we classify sentences that have the same meaning in their common translation in French.

However, the entry for CNN is simply a concatenation of words, so some semantic characteristics of a sentence may be lost.

In this article, an improved CNN model for sentence classification based on term frequency and sentence segmentation is proposed.

Corresponding Author:- Hyacinthe Kouassi Konan
Address:- Ecole Supérieure Africaine Des Technologies d'Information Et De La Communication (ESATIC), Côte d'Ivoire.

First, we code the frequency of each word. Then, sentence segmentation is added during the construction of the word integration matrix. Finally, word frequency and sentence segmentation results are taken as input to our CNN. This article is organized as follows: Section 2 presents neural networks in NLP. Section 3 describes the system we propose for the classification of sentences. Section 4 discusses the experimentation to validate the proposed method. Finally, paragraph 5 concludes the work.

Neural networks in NLP

Language modeling is the first application of neural networks in NLP, and which has been useful in learning distributed representations of words [4]. These interweaving of words guided the new direction of NLP tasks.

Neural networks are therefore increasingly used in NLP. In addition to the above, a class of recursive neural networks and neural tensor networks is proposed for paraphrase detection [5], syntax analysis [6], sentiment analysis [7], completion of the knowledge base [8], answering questions [9], etc.

Much previous work has exploited deep neural networks to model the sentence. The model using an external analysis tree is the recursive neural network (RecNN) [10]. And the recurrent neural network (RNN) [11] is a special case of recursive networks. The RNN can be used as a language model and can also be seen as a sentence model with a linear structure.

CNN is also introduced in sentence modeling. A simple one-layer CNN model [12] combines softmax to perform sentence classification tasks. A CNN typically has one or more convolutional layers and a fully connected final layer, and there will be a pooling layer between every two layers. Combined with the characteristics between CNN and Time-Delay Neural Networks (TDNN) [3], Dynamic Convolutional Neural Networks (DCNN) [13] are used for sentence modeling and perform well on many data sets.

Description of the model

Our proposed CNN-based model is shown in Figure 1. The whole model consists of four main layers: (i) sentence representation, (ii) convolution, (iii) pooling and (iv) logistic regression.

Input

To extract the features at the sentence level input our CNN, we provide two methods as follows.

1) Addition of the frequency of terms (FT): It is necessary to specify which words appear frequently in the sentences. Therefore, the FT is proposed, which is defined as the frequency of the word in the data set. The value of FT is the normalization of the number of terms. In addition, the term frequency of the stop word is set to zero.

2) Segmentation : Each input sentence is divided into several parts in the form of clauses. Thus, more features can be extracted. The length of a sentence decides how many clauses a sentence can have.

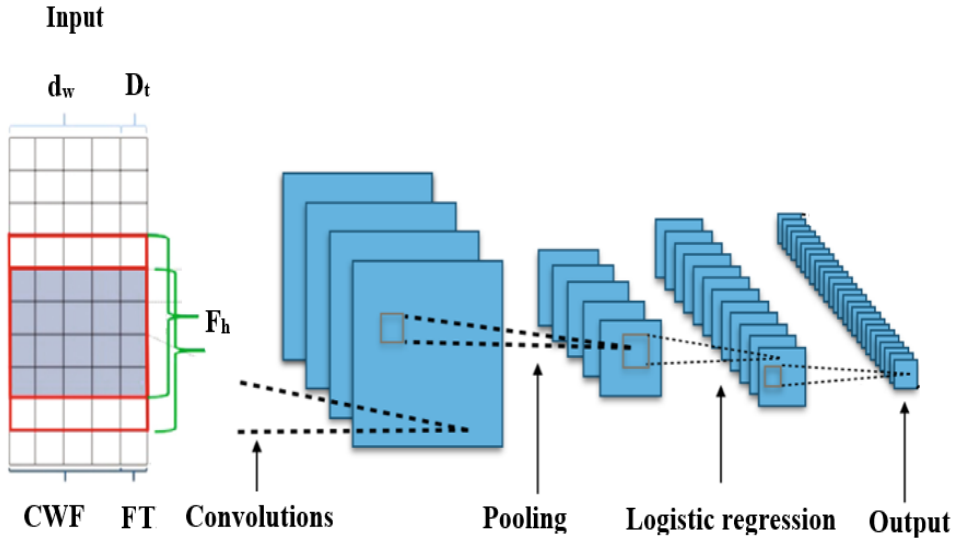

Fig. 1:- Architecture of the model with frequency of terms and segmentation for the classification of sentences.

As shown in Figure 1, the sentence function input module contains two parts:
1. the Context-Word Function (CWF) **[14]**and
2. the FT function.

CWF is the vector of each word transformed by searching for word embeddings **[15]**. The TF function is the term frequency of each word. The entire sentence can be represented by $\boldsymbol{x} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}]$, where $\boldsymbol{x_i}$ is the ith word of the sentence and $\boldsymbol{n}$ is the length of the input sentence.

We define $\boldsymbol{d_t}$ as the width of the integration of the frequency of the term. And We define $\boldsymbol{d_w}$ as the width of the integration of the word. Thus, the entire width of the ith word of the sentence is defined by :

$$\boldsymbol{d} = \boldsymbol{d_t} + \boldsymbol{d_w} \quad (\mathbf{1})$$

Each word of a sentence is replaced by its vector representation, a sentence matrix $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{d}}$ can be obtained. Next, the segmentation step size $\boldsymbol{m}$ is calculated by Equation **(2)**.

$$\boldsymbol{m} = \left\lfloor \frac{\boldsymbol{k}}{\boldsymbol{n}} \right\rfloor \quad (\mathbf{2})$$

Here, k is a fixed parameter. A window of length $\boldsymbol{m}$ is used to divide each phrase from the beginning into $\boldsymbol{w}$ parts, where

$$\boldsymbol{w} = \left\lceil \frac{\boldsymbol{n}}{\boldsymbol{m}} \right\rceil \quad (\mathbf{3})$$

Several zeros are inserted between the $\boldsymbol{w}$ propositions to separate them. The zero number is decided by the height of the max filter window. The number of paddingzerosisdefined as $\boldsymbol{z}$,

$$\boldsymbol{z} = (\boldsymbol{w} - \mathbf{1}) \times \left(\boldsymbol{max}\left(\boldsymbol{F}(\boldsymbol{h})\right) - \mathbf{1}\right) \quad (\mathbf{4})$$

Where $\boldsymbol{F}(\boldsymbol{h})$ denotes the height of the filter windows.

As shown in Figure 1, the red box represents the filter window. When we have to separate two propositions, several zeros must be inserted between them. Only in this way can a convolutional window which contains only one proposition content at any given time in the convolution process be ensured. The number of zeros depends on the height of the max filter window.

As the filter windows of 3, 4, 5 in our experiments are used, at least 4 zeros must be inserted between the propositions as shown in Figure 1 to perform our method. So, the reason why the $(\boldsymbol{max}(\boldsymbol{F}(\boldsymbol{h})) - \mathbf{1})$ is used to denote the number of padding zeros between the propositions is that neither can exactly separate the propositions to get more internal / on-board functionalities nor can it cause redundancy. Thus, the sentence characteristic entry in Figure 1 can be described as a matrix $\boldsymbol{X} \in \mathbb{R}^{(\boldsymbol{n}+\boldsymbol{z})\times\boldsymbol{d}}$

The convolutional layer
The convolutional layer aims to capture the compositional semantics of an entire sentence and compress this valuable semantics into feature maps. Concretely, the following operator is used to obtain another sequence $\boldsymbol{c}$:

$$\boldsymbol{c} = \boldsymbol{f}(\boldsymbol{F}.\boldsymbol{x}_{\boldsymbol{i}:\boldsymbol{i}+\boldsymbol{h}-\mathbf{1}} + \boldsymbol{b}) \quad (\mathbf{5})$$

where a filter $\boldsymbol{F} \in \mathbb{R}^{\boldsymbol{h}\times\boldsymbol{d}}$ and $\boldsymbol{b}$ is a bias term, and $\boldsymbol{f}$ is an activation function, such as $\boldsymbol{tanh}$ or linear unit of rectification (ReLU).

The Pooling layer
To extract the most outstanding features (maximum value) within each feature map, a maximum pooling over time operation [15, 16, 18] on the feature map is applied. The approach takes one trait card as a pool and gets a maximum

value for each trait card. For each filter F, its sequence of c scores is passed through the max function to produce a single number,

$$P_F = max\{c\} = max\{c_1, c_2, \ldots, c_{n-h+1}\} \quad (6)$$

which is used to estimate the n-gram possibility of F appearing in context.

Regularization and Classification

Finally, the pooling scores for each filter are concatenated into a single feature vector o to represent the sentence.

$$o = [P_1, P_2, \ldots P_q] \quad (7)$$

Here, q is the number of filters in the model and P_i is the pooling score of the i-th filter. Next, a dropout [15, 19] is used to prevent co-adaptation of the hidden units by randomly dropping a proportion p of the hidden units during forward and backward propagation. Weights whose l_2 norms exceed a hyperparameter like Kim **[16]**are also rescaled.

Experiences

Data sets

To assess the performance of our proposed method, the summary of the selected data sets is listed in Table 1.Here, a is the number of target classes, l is the average sentence length, N is the size of the dataset, |V| is the size of the vocabulary, |Vpre| is the number of words presented in the set of pretrained word vectors.MR: Film reviews with one sentence per review. Classification consists of detecting positive / negative opinions [17].CR: Customer reviews on various products (camera, MP3 etc.). The task is to predict the positive / negative opinions [18].

Table 1:- Summary of datasets after tokenization.

Dataset	a	l	N	$\|V\|$	$\|Vpre\|$
MR	2	20	10662	18765	16448
CR	2	19	3775	5340	5046

For all datasets we use rectified linear units, filter windows ($F(h)$) of 3, 4, 5 with 100 feature maps each, drop rate (r) of 0.5, a constraint l2 of 3, a parameter k of 300, mini-lot size of 50 and static model. The static model is that all words are pre-trained from a Baule dictionary, and they are kept static during the experiments.

Pre-trained Word Vectors

The word of integration that we used is a Baoulé dictionary which is made up of 4000 words. Vectors have 301 dimensions, including the incorporation of term frequencies and the incorporation of words. If words do not appear in the pretrained words, they are initialized randomly.

Experimental Results and Analysis

The experimental results are listed in Table 2. The first four works of the upper section have good results on the classification of sentences.

The NBSVM method **[19]**uses the Naïve Bayes. SVM and MNB **[19]**use the multinomial Naive Bayes with uni-bigrams to perform sentence classification. The G-Dropout **[20]**uses the Gaussian Dropout formation for classification and Tree-CRF **[21]**presents a method based on a dependency tree using conditional random fields with hidden variables. The following three jobs in the middle section are advanced methods. CNN-static **[12]**is a model with pretrained vectors and all words are kept static and only other parameters are learned.

Table 2:- Results of our CNN models against other methods.

Model	MR	CR
NBSVM **[19]**	79.4	81.9
MNB **[19]**	79.3	80.2
G-Dropout **[20]**	79.3	82.2
Tree-CRF **[21]**	77.5	81.5
CNN-static **[1]**	81.3	84.8
CNN-non-static **[12]**	81.5	84.5

CNN-multichannel **[12]**	81.3	85.3
CNN-TF	81.7	85.5
CNN-TF-Segmentation	81.9	85.9

CNN-non-static **[12]**is the same as CNN-static but the pretrained vectors are refined for each task. CNN-multichannel [12] uses two sets of word vectors and each set is treated as a channel. The jobs in the bottom section are our methods.

The results of our methods are significantly better than those of the above methods. Note that the static CNN model which does not allow updating of input embeddings during training is used in our experiment to explore the performance of our methods which introduced TF and sentence segmentation.

When using CNN-TF, it can get better results than most other methods except non-static CNN in MR and multichannel CNN in CR. One possible reason is that TF plays an important role in extracting the semantic characteristics of the sentence. TF represents the frequency of a word that determines whether that word is needed or not. If a word appears frequently (except stop words) in a data set, its weights should be higher.

When using CNN-TF segmentation, it can achieve the best results in MR and CR. These results show that internal / edge features can be extracted by sentence segmentation. These characteristics play an important role in the classification of sentences. Thus, the frequency of the terms and the internal / edge characteristics are necessary for the classification of the sentences.

Conclusion:-

In this article, an improved CNN method for sentence classification is proposed. On the one hand, the term frequency is introduced to be an important characteristic. The higher the frequency of terms in a word, the more important the word.

On the other hand, sentence segmentation is introduced to produce more sentence clauses, and sentence clauses are presented to learn more about internal characteristics and marginal characteristics and characteristics are very useful for classification. The effectiveness of our approach is verified by applying it to the classification of sentences on two sets of reference data: a film review dataset called MR and a dataset of customer reviews called CR. The experimental results show that our method gives satisfactory results.

Reference:-

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Proceedings of the 26th Annual Conference on Neural Information Processing Systems, pp. 1097–1105 (2012)
2. Graves, A., Mohamed, A.R., Hinton, G.E.: Speech recognition with deep recurrent neural networks. In: Proceedings of the 38th International Conference on Acoustics, Speech and Signal Processing, pp. 6645–6649. IEEE Press, New York (2013)
3. Collobert, R., Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning, pp. 160–167. ACM, New York (2008)
4. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and of words and phrases and their compostionality. In: Proceedings of the 27th Annual Conference on Neural Information Processing Systems, pp. 3111–3119 (2013)
5. Socher, R., Pennington, J., Huang, E.H., Ng, A.Y., Manning, C.D.: Semisupervised recursive autoencoders for predicting sentiment distributions. In: Proceedings of the 8th Conference on Empirical Methods in Natural Language Processing, pp. 151–161. ACL Press, Stroudsburg (2011)
6. Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 10th Conference on Empirical Methods in Natural Language Processing, pp. 1631–1642. ACL Press, Stroudsburg (2013)
7. Socher, R., Bauer, J., Manning, C.D., Ng, A.Y.: Parsing with compositional vector grammars. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pp. 455–465. ACL Press, Stroudsburg (2013)

8. Socher, R., Chen, D., Manning, C.D., Ng, A.Y.: Reasoning with neural tensor networks for knowledge base completion. In: Proceedings of the 27th Annual Conference on Neural Information Processing Systems, pp. 464–469 (2013)
9. Iyyer, M., Boyd-Graber, J., Claudino, L., Socher, R., Iii, H.D.: A neural network for factoid question answering over paragraphs. In: Proceedings of the 11th Conference on Empirical Methods in Natural Language Processing, pp. 633–644. ACL Press, Stroudsburg (2014)
10. Socher, R., Huang, E.H., Pennington, J., Ng, A.Y., Manning, C.D.: Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In: Proceedings of the 25th Annual Conference on Neural Information Processing Systems, pp. 801–809 (2011)
11. Mikolov, T., Kombrink, S., Burget, L., Cernocky, J.H.: Extensions of recurrent neural network language model. In: Proceedings of the 36th IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 5528–5531. IEEE Press, New York (2011)
12. Kim, Y.: Convolutional neural networks for sentence classification. In: 11th Conference on Empirical Methods in Natural Language Processing, pp. 1746–1751. ACL Press, Stroudsburg (2014)
13. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 655–665. ACL Press, Stroudsburg (2014)
14. Chen, Y., Xu, L., Liu, K., Zeng, D., Zhao, J.: Event extraction via dynamic multipooling convolutional neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, pp. 167–176. ACL Press, Stroudsburg (2015)
15. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. 12(1), 2493–2537 (2011)
16. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. R.: Improving Neural Networks by Preventing Co-adaptation of Feature Detectors. CoRR, abs/1207.0580 (2012)
17. Pang, B., Lee, L.: Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, pp. 115–124. ACL Press, Stroudsburg (2005)
18. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177. ACM, New York (2004)
19. Wang, S., Manning, C.D.: Baselines and bigrams: simple, good sentiment and topic classification. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pp. 90–94. ACL Press, Stroudsburg (2012)
20. Wang, S., Manning, C.D.: Fast dropout training. In: Proceedings of the 30th International Conference on Machine Learning, pp. 118–126 (2013)
21. Nakagawa, T., Inui, K., Kurohashi, S.: Dependency tree-based sentiment classification using CRFs with hidden variables. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 786–794 (2010).