



A Dynamic Convolutional Neural Network Approach for Legal Text Classification

Eya Hammami¹(✉), Rim Faiz², and Imen Akermi¹

¹ LARODEC Laboratory, University of Tunis, Tunis, Tunisia
imen.akermi@isg.rnu.tn

² LARODEC Laboratory, University of Carthage, Carthage, Tunisia
rim.faiz@ihec.rnu.tn

Abstract. The Amount of legal information that is being produced on a daily basis in courts is increasing enormously. The processing of such data has been receiving considerable attention thanks to their availability in an electronic form and the progress made in Artificial Intelligence application. Indeed, deep learning has shown promising results when used in the field of natural language processing (NLP). Neural Networks such as convolutional neural networks and recurrent neural network have been used for different NLP tasks like information retrieval, sentiment analysis and document classification. In this work, we propose a Neural Network based model with a dynamic input length for French legal text classification. The proposed approach, tested over real legal cases, outperforms baseline methods.

Keywords: Natural language processing · Document categorization · Legal domain · Artificial intelligence

1 Introduction

The continued application of computational intelligence in legal domain has drawn a lot of attention in the last few decades. With the increased availability of legal text in digital form a wide variety of applications, including summarization [1], reasoning, classification [2], translation, text analytics, and others have been addressed within the legal domain. In this paper we particularly tackle text classification. Indeed, there are several applications that require partitioning natural language data into groups, e.g. classifying opinions retrieved from social media sites, or filtering spam emails, etc. In this work, we assert that law professionals would considerably gain advantage from the potential supplied by machine learning. This is especially the case for law professionals who have to take complicated decisions regarding several aspects of a given case. Given data accessibility and machine learning techniques, it is possible to train text categorization systems to predict some of these decisions. Such systems can act as a

ICIKS 2021.

decision support system for law professionals. Several approaches have been proposed for text classification to mention, Naive Bayes classifier, Support Vector Machine, Logistic Regression, and most recently deep learning methods such as Convolutional Neural Network (CNN) [3,4], Recurrent Neural Network (RNN) and Long-Short Term Memory (LSTM) [5]. Most of these approaches are not particularly designed for the legal domain and are usually trained with English text, which make them not appropriate to be used for French text and particularly legal French text. Indeed, French is a language with a richer morphology and a more flexible word order, that requires more preprocessing to achieve good accuracy results and capture the hidden semantics specially when dealing with legal texts. In this paper, we propose NN-based model with dynamic input length layer to process French legal data. We also present a comparative study between the proposed approach and several baseline models. This paper is organized as follows: we present in Sect. 2 a literature review that examines the different approaches for text classification. In Sect. 3, we describe our proposed model. Finally, experiments and the deployment part are presented in Sect. 4.

2 Related Work

This section presents a brief discussion on the text classification task and on the application of deep learning to legal domain which includes various models developed for retrieving and classifying relevant legal text. Text classification is a necessary task in Natural Language Processing. Linear classifiers were frequently used for text classification [6,7]. According to [8] these linear models could scale up to a very huge dataset rapidly with a proper rank constraint and a fast loss approximation. Deep learning methods, such as recurrent neural networks (RNN) and Long Short Term Memory (LSTM) have been widely used in language modeling. Those methods are adapted to natural language processing because of their ability to extract features from sequential data, to mention the Convolutional Neural Network (CNN) which is usually used for computer vision tasks like in [9–11]. This model has been adopted in NLP for the first time in [12]. In this work the authors presented a new global max-pooling operation, which was revealed to be efficient for text, as an alternative to the conventional local max-pooling of the original LeNet architecture [13]. Furthermore, they suggested to transfer task-specific information by co-training different deep models on many tasks.

Inspired by the original work of [9,12] introduced a simpler architecture with modifications consisting of fixed pre-training word2vec embeddings. They proceed that both multitask learning and semi-supervised learning enhance the generalization of the shared tasks, resulting in state-of-the-art-performance. Moreover, in [14], the authors demonstrated that this model can actually achieve state-of-the-art performances on many small datasets. Dynamic Convolutional Neural Network (DCNN) is a type of CNN which is introduced by [15]. Their approach outperforms other methods on sentiment classification. They use a new pooling layer called a dynamic K-max pooling. This dynamic k-max pooling is

a generalization of the max pooling operator, which computes a new adapted K value for each iteration. Thus, their network can read any length of an input. Character-level Convolutional Neural Network (Char-CNN) which is introduced by [10] also yields better results than other methods on sentiment analysis and text classification. In the same context, [16] shows that a character-based embedding in CNN is an effective and efficient technique for sentiment analysis that uses less learnable parameters in feature representation. Their proposed method performs sentiment normalization and classification for unstructured sentences. A new Char-CNN model proposed by [17] and inspired from the work presented in [10], allows any length of input by employing k-max pooling before a fully connected layer to categorize Thai news from a newspaper. Furthermore, the work in [18] presented a character aware neural language model by combining a CNN on character embeddings with a Highway-LSTM on subsequent layers. Their results suggest that on many languages, character inputs are relevant for language modeling. In addition, [19] analyzed a multiplicative LSTM (mLSTM) on character embeddings and found out that a basic logistic regression learned on this representation can reach state-of-the art results on the Sentiment TreeBank dataset [20] with a few hundred labeled examples. We have noticed a rather small body of previous works about automatic text classification for legal documents. For example, support vector machines (SVMs) have been used to classify legal documents like legal docket entries in [21]. The authors developed simple heuristics to address the conjunctive and disjunctive errors of classifiers and improve the performance of the SVMs. Based on the prescience gained from their experiments, they also developed a simple propositional logic based classifier using hand labeled features, that addresses both types of errors simultaneously. They proved that this simple, approach outperforms all existing state-of-the-art ML models, with statistically significant gains. A mean probability ensemble system combining the output of multiple SVM classifiers to classify French legal texts, was also developed by [22]. They reported accuracy scores of 98% for predicting a case ruling, 96% for predicting the law area of a case, and 87.07% on estimating the date of a ruling. A preliminary study addressing deep learning for text classification in legal documents was proposed in [23]. They compared deep learning results with results obtained using SVM algorithm on four datasets of real legal documents. They demonstrated that CNN present better accuracy score with a training dataset of larger size and can be improved for text classification in the legal domain. Neural Networks such as CNN, LSTM and RNN have also been used for classifying English legal court opinions of Washington University School of Law Supreme Court Database (SCDB) in [24]. The authors compared the machine learning algorithms with several Neural Networks systems and they found out that CNN combined with Word2vec performed better compared to the other approaches and gave an accuracy around 72.7%. Based on the Brazilian Court System representing the biggest judiciary system in the world, and receiving an extremely high number of lawsuits every day, the work in [25] presented a CNN based approach that helps analyse and classify these cases, in order to be associated to relevant tags and allocated to the right team. The obtained results

are very promising. However, most of the mentioned approaches are generally based on the CNN model and usually use a static input length. Therefore, we propose to experiment this model with a dynamic input length on French legal data. Experiments on real datasets highlight the relevance of our approach and open up many perspectives.

3 Proposed Model

The architecture of our proposed model, shown in Fig. 1, is based on the CNN model [24], characterized by a max pooling layer also called temporal max pooling, which is a method for down sampling data by utilizing a gliding window on a row of data and choosing a cell which includes a maximum value to be moved to the next layer. It carries out an operation on 1D CNN and it is calculated by the following formula (1) [17]:

$$P_{r,c} = \max_{j=1}^s M_{r,s(c-1)+j} \quad (1)$$

where:

- M is an input matrix with a dimension of $n \times l$
- s is a pooling size
- P is an output matrix with a dimension of $n \times \frac{l}{s}$
- c is a column cell of matrix P
- r is a row cell of matrix P

It is within this pooling layer that we try to experiment pooling layer, thus using the k-max pooling layer rather than the max-pooling layer. In fact the k-max pooling operation enables to pool the k maximum active features in P, also, it keeps the order of the features, but it is insensitive to their accurate positions. It can then detect more delicately the number of times where the feature is activated in P. The k-max pooling operator is used in the network after the highest convolutional layer. This allows the input to the fully connected layers to be separate from the length of the input sentence. Additionally, in the middle of the convolutional layers, the pooling parameter k is not fixed, but is selected in a dynamic way to enable a sleek extraction of a longer-range and higher order features [15]. This dynamic pooling layer is calculated by the following formula (2) [17]:

$$P_{r,*} = k \max_{j=1}^l M_{r,j} \quad (2)$$

where:

- M is an input matrix with a dimension of $n \times l$
- K is an integer value
- P is an output matrix with a dimension of $n \times k$
- $*$ shows that all columns in a row are calculated together
- r is a row cell of matrix P

The main difference between these two types of pooling layer consists in the use of a gliding window. Max pooling is a method for down sampling data by utilizing a gliding window on a row of data and choosing a cell which includes a maximum value to be moved to the next layer [17]. Differently, k-max pooling doesn't have a window, but it has a choosing operation which carries out all data in a row. Then, top k cells which have maximum value are chosen to be utilized in the up-coming layer [17]. By applying K-max pooling in a convolutional neural network, as we propose, we can definitely have a matrix which is able to fit into a fully connected layer regardless of the length of an input. Figure 1 illustrates in details our proposed method.

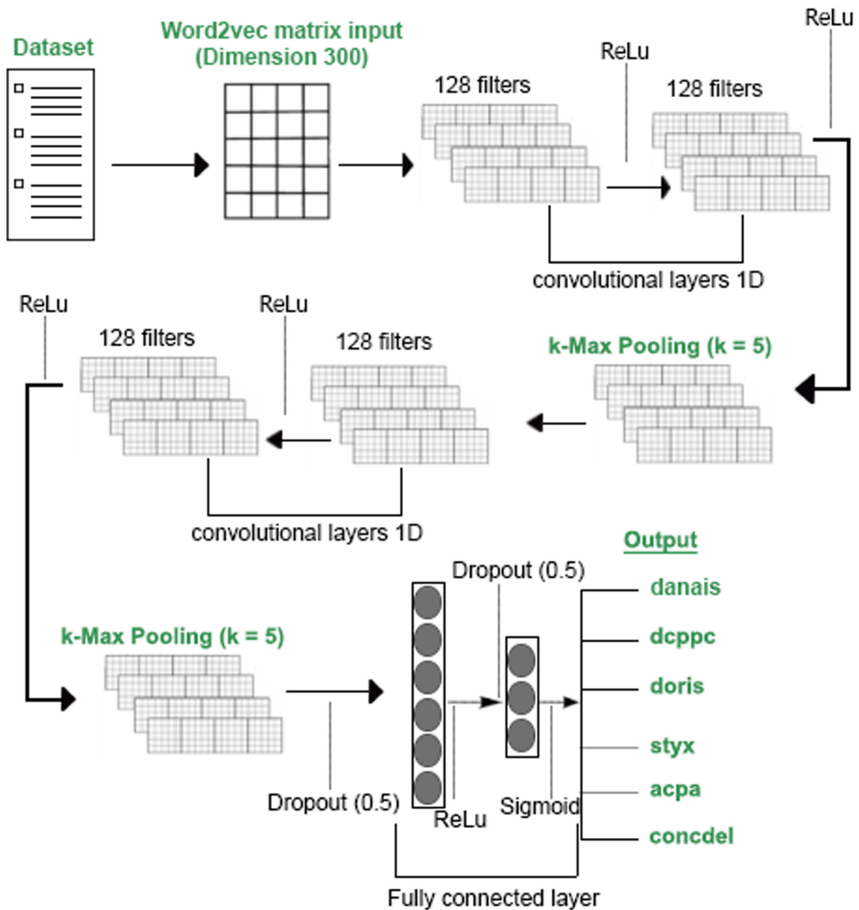


Fig. 1. Proposed architecture.

On the convolutional and pooling layers, the data length belongs to the input. Whereas, after the k-max pooling layer, the data length in each document is

coequal. Thus, our neural network classification model is a little bit similar to the one introduced by [9], but we modified the layers, by adding other layers and modifying some of the original hyperparameters in order to obtain a better performed text categorization model. Our model first makes an embedding layer using word2vec as a pre-trained word embedding, and next makes a matrix of documents represented by 300-dimensional word embedding. As we all know when employing machine learning methods in NLP, most of the studies use 200 or 300 dimensional vectors, but 300-dimensional embedding carry more information and this, therefore, is considered to produce better performance results. Then, we incorporate the following sets of parameters: A dropout of 0.5, because it helps to change the concept of learning all the weights together to learning a fraction of the weights in the network in each training iteration; a convolution layer of 128 filters with a filter size of 3, according to the literature, we set the k value to 5. We also add a dense layer consisting of 128 units between two dropouts of 0.5 to prevent overfitting. Finally, the last layer (output layer) is a dense layer with a size of 6 equal to the number of labels (categories) in our dataset.

4 Experiments and Results

We present in this section the experimental results of our approach compared to the different methods used in the literature. We use the accuracy and the F_1 scores to evaluate these models.

4.1 Dataset

We trained and tested our model on a French legal dataset collected from data.gouv.fr¹. It is a documentary collection of lawsuits from French courts. The dataset includes 2000 documents (txt files). The following figure (Fig. 2) presents a sample of this dataset. This dataset is organized into 6 categories whose denomination were carried out by a legal expert (see Table 1). The number of documents was limited because the documents annotation is done manually and exclusively by legal experts. Work is underway to try to expand the training corpus. After processing, the vocabulary size is 3794659. We randomly divide it into training and test set, with 80% and 20% split.

4.2 Pre-processing

Our model first removes special characters like punctuation, stopwords, numbers and whitespaces. The removal of these special characters will allow us to have classes that are representative of the words that are recurring in our documents. Second, we proceed with lemmatization by using TreetaggerWrapper module and removing named entities after recognize them using French Spacy and NLTK

¹ www.data.gouv.fr/fr/.


 CAAIX1416771 - Bloc-notes
 Fichier Edition Format Affichage Aide
 Cour d'appel, Aix-en-Provence, 10e chambre, 4 Mai 2016 - n° 14/16771
 Cour d'appel
 Aix-en-Provence
 10e chambre
 4 Mai 2016
 Numéro de rôle: 14/16771
 Numéro: 2016/204
 X / Y
 Contentieux Judiciaire
 COUR D'APPEL D'AIX-EN-PROVENCE
 10e Chambre
 ARRÊT AU FOND
 DU 04 MAI 2016
 N° 2016/204
 Rôle N° 14/16771
 Pasquale D.
 C/
 Thierry H.
 CAISSE PRIMAIRE D'ASSURANCE MALADIE DES ALPES MARITIMES
 SA FCA MOTOR VILLAGE FRANCE
 Grosse délivrée
 le:
 à:
 Me Franck G.
 SCP M. -V.-M.,
 Me Vincent P.
 Décision déferée à la Cour:
 Jugement du Tribunal d'Instance de NICE en date du 17 Juillet 2014 enregistré au
 répertoire général sous le n° 12-000238.
 APPELANT
 Monsieur Pasquale D.
 né le 28 Mars 1970 à [...], de nationalité Française,
 demeurant [...]
 représenté par Me Franck G., avocat au barreau de GRASSE
 INTIMES
 Monsieur Thierry H.
 <

Fig. 2. Document sample.

Table 1. Predefined categories.

Number of documents	Label
298	DANAIS
501	DCPPC
159	DORIS
160	STYX
582	CONCDEL
300	ACPA

modules to allow a more accurate interpretation of the data. We chose to perform lemmatization rather than stemming because lemmatization considers the context and converts the word to its meaningful base form, whereas stemming just removes the last few characters, often leading to incorrect meanings and spelling errors. Finally, each word in the corpus is mapped to a word2vec vector before being fed into the convolutional neural network for categorization.

4.3 Experiments

In this paper, we use Accuracy as a measure of evaluation in order to determine the degree of predictions that the models was able to guess correctly. It is calculated like the following (3):

$$Accuracy = \frac{\text{number of correctly classified documents}}{\text{total number of classified documents}} \quad (3)$$

We also consider the F_1 score (4) which is a metric that combines both Precision and Recall using the Harmonic mean. In this work, our classification problem based on imbalanced class distribution, thus F_1 score is a better metric to evaluate our model on. $F_{1,i}$ refers to F_1 of class i , C is the number of categories:

$$F_1 = \frac{\sum_{i=1}^C F_{1,i}}{|C|} \quad (4)$$

Where:

$$F_{1,i} = 2 \cdot \frac{\text{precision}_i \cdot \text{recall}_i}{\text{precision}_i + \text{recall}_i} \quad (5)$$

Along with our CNN K-max pooling approach, we experimented two other CNN based models: CNN max-pooling and CNN global max-pooling. In the CNN with max pooling, we use the same hyperparameters as the CNN with global max pooling, but we change the pooling size to 3. The implementation of these three architectures is done using Keras which allows users to choose whether the models they build are running on Theano or TensorFlow. In our case the models run on TensorFlow.

Regularization of Hyperparameters: In our experiments, we tested our model with a set of various hyperparameters. The model performed best when using 128 filters for each of the convolutional layers. In addition, each of the models is adjusted with a dropout [27], which works by “dropping out” a proportion p of hidden units throughout training. We found out that a dropout of 0.5 and a batch size of 256 worked best for our CNNs models, along with the Adam optimizer [26].

Results of the First Experiment ($K = 5$): As shown in Table 2, CNN with max pooling performs better. It can achieve an accuracy of 84,46 %, which outperforms the CNN with k-max pooling (our proposed approach) and the CNN with global max pooling. We think that this is maybe due to the limits number of documents in the dataset.

Table 2. Results of the first experiments where k value set to 5.

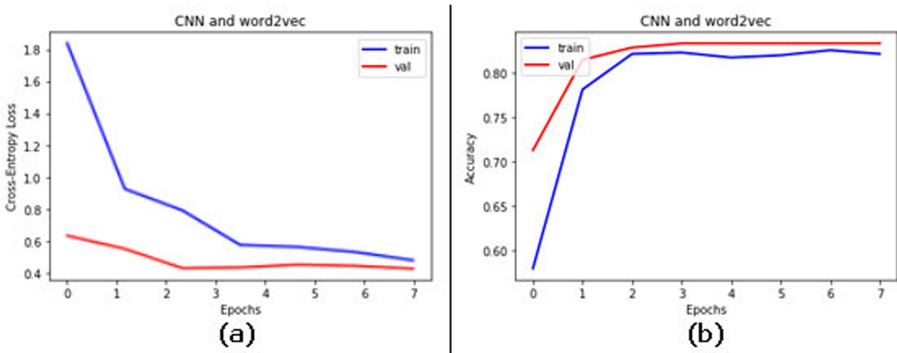
Method	Accuracy (%)	F1 (%)
CNN with k-max pooling	80.35	80.20
CNN with max pooling	84,46	84.46
CNN with global max pooling	81,94	82.10

Results of the Second Experiment ($K = 3$): Now we decrease the k value to 4 then to 3. The purpose of this second experiment was explore if we could get a better accuracy when varying K with the proposed K-max pooling approach. As shown in Table 3, our model outperforms the other models with 84.71% accuracy when K is set to 3.

Table 3. Results of the second experiments where k value set to 3 which outperforms the other models.

Method	Accuracy (%)	F1 (%)
CNN with k-max pooling ($K = 3$)	84.71	84.80
CNN with k-max pooling ($K = 4$)	83,32	83.11
CNN with max pooling	84,46	84.46
CNN with global max pooling	81,94	82.10

As follows we present the two plots of accuracy and Cross Entropy for this second experiment: Fig. 3 shows the regression of the Cross Entropy and the evolution of the accuracy for CNN with k-max pooling (where $k = 3$) according to the number of epochs for both training and test sets. The red curve corresponds to the validation and the blue curve corresponds to the training. In this two graphics we notice that the line plot is well converged for the two curves and gives no sign of over or under fitting.

**Fig. 3.** Line plot of Cross Entropy Loss over Training Epochs.

Comparison with Baseline Methods: We also compared the CNN based models to three traditional methods: Naive Bayes Classifier (with TF-IDF) [28], Word2vec embedding with Logistic Regression [29] and SVM [21]. The results are shown in Table 4 as we can notice, CNN with k-max pooling outperforms non NN based models.

Table 4. Results of comparison with baseline methods.

Method	Accuracy (%)	F1 (%)
CNN with k-max pooling	84.71	84.80
CNN with max pooling	84,46	84.46
CNN with global max pooling	81,94	82.10
Naive Bayes classifier	41,91	42,00
Word2vec and Logistic Regression	80,88	80,01
SVM	79,98	79,51

Deployment: We developed a small desktop application based on our proposed method, which is designed for law professionals to allow them to categorize automatically textual data.

Figure 4 presents a screenshot of the home interface and Fig. 5 shows how they can easily load a simple French legal txt file to predict its classification according to predefined categories (see Fig. 6).

However, We are currently working on enhancing this application in order to integrate more functionalities that can help law professionals with heavy manual tasks.

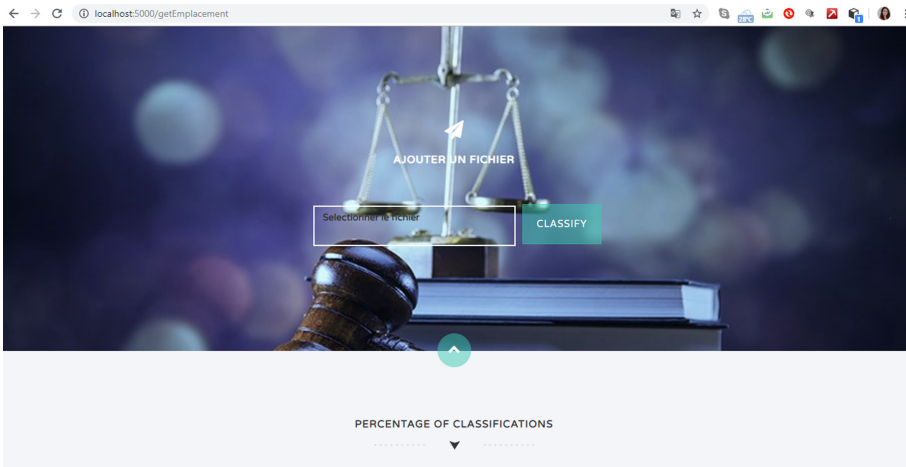


Fig. 4. The application Home page.

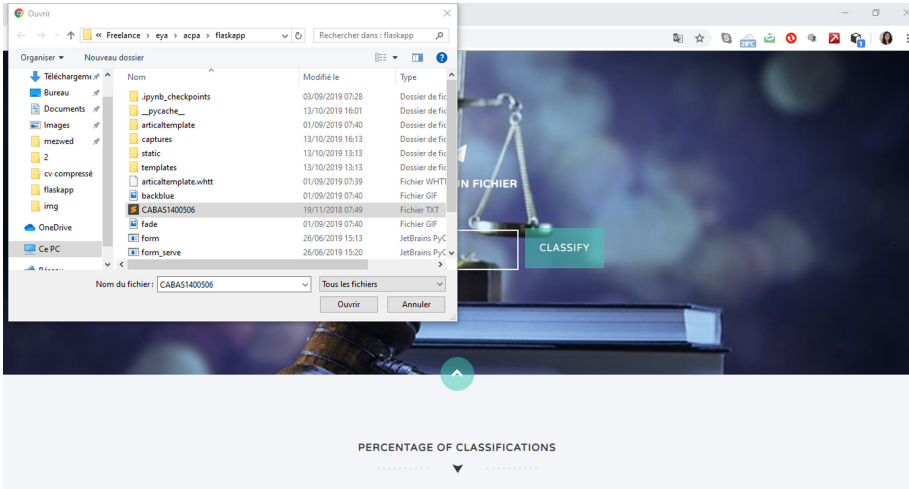


Fig. 5. Load a txt file.

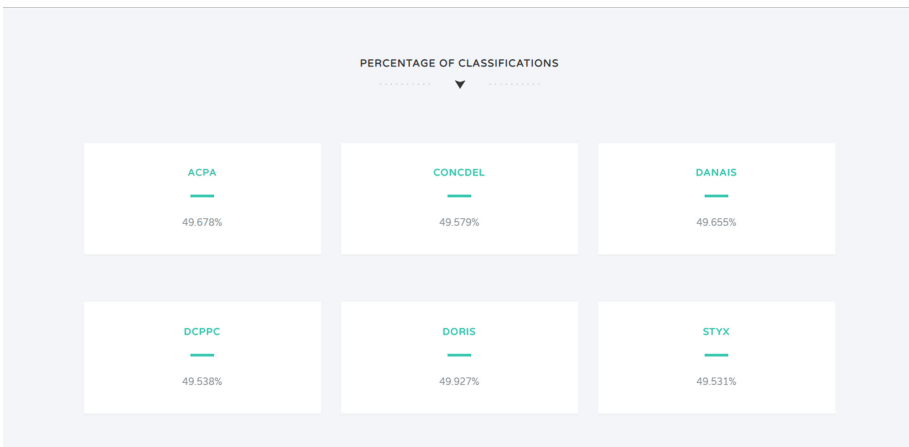


Fig. 6. Percentages of a txt file classification.

5 Discussion

Dynamic max pooling [17], usually proved to perform much better compared to classic max pooling and other baseline methods. But in our first experiment, the static max pooling outperforms all other methods with an accuracy of 84,46%. Then in our second experiment, we adjusted the K value of the Dynamic max pooling to 3, as a result the Dynamic max pooling gets better accuracy result of 84,71%. By the way, it outperforms all other methods. In this work we considered that with dynamic k-max pooling, the value of k depends on the input shape. The idea is that longer sentences can have more max values (higher k). But in our

case, the sentence's length that we have, are not enough to set higher K value. We considered also that the words contained in the pre-trained word embedding may not capture the specificity of languages in legal domain. Therefore, we think maybe for these reasons our results may not be very optimal especially for the first experiment.

6 Conclusion

In this paper, we addressed the use of CNN with dynamic input length for French legal data classification. Our suggested approach, which can process a longer input length, outperforms the original model with a fixed input length in terms of accuracy.

A number of interesting future works have to be mentioned:

Firstly, we plan to re-adjust the network architecture, so it can better capture the characteristics of our French legal data.

Secondly, we should test our proposed approach on new datasets to validate its performance.

Finally, We think that we can extend our reflections to the categorization of hand written documents and not be limited to electronic versions.

Acknowledgments. This paper has been done under the contract PREMATTAJ 2017–2019 of the Occitanie region which is greatly acknowledged. The decisions used in this paper have been annotated by Professor Guillaume Zambrano of the University of Nîmes.

References

1. Doan, T.M., Jacquenet, F., Largeron, C., Bernard, M.: A study of text summarization techniques for generating meeting minutes. In: Dalpiaz, F., Zdravkovic, J., Loucopoulos, P. (eds.) RCIS 2020. LNBIP, vol. 385, pp. 522–528. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50316-1_33
2. Stead, C., Smith, S., Busch, P., Vatanasakdakul, S.: Towards an academic abstract sentence classification system. In: Dalpiaz, F., Zdravkovic, J., Loucopoulos, P. (eds.) RCIS 2020. LNBIP, vol. 385, pp. 562–568. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50316-1_39
3. Li, Y., Hao, Z.B., Lei, H.: Survey of convolutional neural network. *J. Comput. Appl.* **36**, 2508–2515 (2016)
4. Albawi, S., Mohammed, T.A., Al-Zawi, S.: Understanding of a convolutional neural network. In: International Conference on Engineering and Technology, ICET, pp. 1–6 (2017)
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997)
6. Joachims, T.: Text categorization with Support Vector Machines: learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998). <https://doi.org/10.1007/BFb0026683>

7. McCallum, A., Nigam, K.: A comparison of event models for Naive Bayes text classification. In: AAAI-98 Workshop on Learning for Text Categorization, pp. 41–48 (1998)
8. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, pp. 427–431 (2017)
9. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751 (2014)
10. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Proceedings of the 29th Conference on Neural Information Processing Systems, NIPS 2015. Advances in Neural Information Processing Systems, pp. 649–657 (2015)
11. Conneau, A., Schwenk, H., Barrault, L., Lecun, Y.: Very deep convolutional networks for text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, pp. 1107–1116 (2017)
12. 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, ICML, pp. 160–167 (2008)
13. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE, pp. 2278–2324 (1998)
14. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: International Conference on Learning Representations (2013)
15. 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 (2014)
16. Arora, M., Kansal, V.: Character level embedding with deep convolutional neural network for text normalization of unstructured data for Twitter sentiment analysis. *Soc. Netw. Anal. Min.* **9**(1), 1–14 (2019). <https://doi.org/10.1007/s13278-019-0557-y>
17. Koomsubha, T., Vateekul, P.: A character-level convolutional neural network with dynamic input length for Thai text categorization. In: Proceedings of the 9th International Conference on Knowledge and Smart Technology, KST, pp. 101–105 (2017)
18. Kim, Y., Jernite, Y., Sontag, D., Rush, A.M.: Character-aware neural language models. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (2016)
19. Radford, A., Jozefowicz, R., Sutskever, I.: Learning to generate reviews and discovering sentiment. In: International Conference on Learning Representations, ICLR (2018)
20. Socher, R., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, pp. 1631–1642 (2013)
21. Nallapati, R., Manning, C.D.: Legal docket classification: where machine learning stumbles. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pp. 438–446 (2008)
22. Sulea, O.M., Zampieri, M., Malmasi, S., Vela, M., Dinu, L.P., van Genabith, J.: Exploring the use of text classification in the legal domain. In: Proceedings of 2nd Workshop on Automated Semantic Analysis of Information in Legal Texts, ASAIL (2017)

23. Wei, F., Qin, H., Ye, S., Zhao, H.: Empirical study of deep learning for text classification in legal document review. In: International Conference on Big Data, pp. 3317–3320 (2018)
24. Undavia, S., Meyers, A., Ortega, J.E.: A comparative study of classifying legal documents with neural networks. In: Federated Conference on Computer Science and Information Systems, FedCSIS, pp. 515–522 (2018)
25. Da Silva, N.C., et al.: Document type classification for Brazil’s supreme court using a convolutional neural network. In: Proceedings of the Tenth International Conference on Forensic Computer Science and Cyber Law-ICoFCS, pp. 7–11 (2018)
26. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations (2015)
27. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from over fitting. *J. Mach. Learn. Res.* JMLR 1929–1958 (2014)
28. Yoo, J.-Y., Yang, D.: Classification scheme of unstructured text document using TF-IDF and Naive Bayes classifier. *The Journal of Machine Learning Research, Proceedings of 3rd International Conference on Computer and Computing Science, COMCOMS* (2015)
29. Pranckevičius, T., Marcinkevičius, V.: Comparison of Naive Bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. *Baltic J. Mod. Comput.* **5**, 221 (2017)