A Bayesian CNN-LSTM Model for Sentiment Analysis in Massive Open Online Courses MOOCs

https://doi.org/10.3991/ijet.v16i23.24457

Khaoula Mrhar[1](✉), Lamia Benhiba[1], Samir Bourekkache[2], Mounia Abik[1]
[1] Mohammed V University in Rabat, Morocco
[2] University of Biskra, Biskra, Algeria
Khaoula_mrhar@um5.ac.ma

Abstract—Massive Open Online Courses (MOOCs) are increasingly used by learners to acquire knowledge and develop new skills. MOOCs provide a trove of data that can be leveraged to better assist learners, including behavioral data from built-in collaborative tools such as discussion boards and course wikis. Data tracing social interactions among learners are especially interesting as their analyses help improve MOOCs' effectiveness. We particularly perform sentiment analysis on such data to predict learners at risk of dropping out, measure the success of the MOOC, and personalize the MOOC according to a learner's behavior and detected emotions. In this paper, we propose a novel approach to sentiment analysis that combines the advantages of the deep learning architectures CNN and LSTM. To avoid highly uncertain predictions, we utilize a Bayesian neural network (BNN) model to quantify uncertainty within the sentiment analysis task. Our empirical results indicate that: 1) The Bayesian CNN-LSTM model provides interesting performance compared to other models (CNN-LSTM, CNN, LSTM) in terms of accuracy, precision, recall, and F1-Score; and 2) there is a high correlation between the sentiment in forum posts and the dropout rate in MOOCs.

Keywords—MOOCs, sentiment analysis, deep learning

1 Introduction

Massive open online courses MOOCs have been growing exponentially over recent years due to the flexibility of access and the variety of covered topics [1]. Indeed, between late 2011 and 2016, over 58 million learners were enrolled in 6850 courses created by more than 700 institutions [2]. Collaborative learning is one of the advantages of MOOCs over traditional e-learning platforms. It aims to promote learners' individual cognition, group cognition, and community cognition through the use of several tools [3] like discussion boards, wikis and blogs. These tools provide learners with a space to share knowledge, exchange ideas, seek help, and express their feelings. In fact, interactions among learners can help sustain their interest and engagement, and improve their learning performance overall.

Tracing social interactions on MOOCs by analyzing discussion boards is instrumental to investigating learners' opinions [4] and ultimately improving MOOCs effectiveness. Firstly, it allows us to analyze learners' sentiments to predict learners at risk of dropping out, and classify learners in different groups according to their personality. Moreover, it provides the opportunity of analyzing learners' satisfaction in order to measure MOOCs success and build a personalized recommender system for educational resources. Sentiment analysis has been used in diverse fields for various purposes, such as mining opinions on social media, analyzing reviews and opinions in commerce context, or investigating movies reviews [5]. Sentiment analysis remains however a challenging task because the human language is incredibly diverse, complex, and constantly changing. Thereby, teaching a machine how to analyze the grammatical nuances and cultural variations, and understand the emotions, sentiments, and meanings that occur in reviews provided by learners, inter alia, is a difficult process.

In the last decade, deep learning has emerged as a powerful machine learning technique that produces state-of-art results in natural language processing and particularly sentiment classification. Deep learning models are very suitable for sentiment analysis due to their automatic learning capability and their ability to handle huge amount of data. Different deep learning models were used for this task such as convolutional neural networks or long short-term memory (LSTM). Traditional deep learning techniques are however based on real-valued deterministic models and thereby aren't well designed to model the uncertainty associated with the predictions they make [6]. One way to capture this uncertainty is to apply Bayesian deep learning approaches which offer a practical framework for quantifying confidence levels or uncertainties of deep learning predictions. The use of uncertainty in MOOCs is crucial because mispredictions may affect the MOOC's effectiveness and impact learners negatively.

In this paper we address sentiment classification in MOOCs to help generate a pedagogical strategy that retains learners at risk of dropping out. Our study has hence two general research objectives:

Propose a Bayesian CNN-LSTM model for sentiment analysis in MOOCs. Introducing uncertainty in our predictions will allow us to create an adaptive and personalized pedagogical approach for learners that will help them complete their courses efficiently.

Explore the importance of the sentiment analysis model in MOOCs by investigating the link between sentiment analysis of learners' forum posts and dropout in MOOCs.

The remainder of this paper is organized as follows. The second section briefly reviews the state of art in sentiment analysis using deep learning techniques. In the third section, we briefly review the two architectures of deep learning CNN and LSTM and the Bayesian neural network model. The fourth section presents the proposed Bayesian CNN-LSTM model. It details the experiments that evaluate the effectiveness of our method and reports the analysis of results. Finally, we conclude with a summary and some perspectives for future work.

2 Related work

Sentiment analysis is a Natural Language Processing (NLP) research area that emerged in the past decade with the aim to automatically extract and classify opinions in text. With applications spanning various areas (from marketing strategies, recommendation systems, to financial forecasting), sentiment analysis in the social media context is facing several challenges. Firstly, there are huge amounts of data available with varying quality. Secondly, messages on social networks are written in human language which is highly complex and oftentimes ambiguous. In effect, the same sentiment can be used to express two different ideas in two different contexts. To address these challenges, researchers are increasingly turning to Deep Learning algorithms as they achieved significant results in various NLP tasks, such as automatic summarization, machine translation, question answering etc. [7].

Several novel Deep Learning architectures were proposed to capture sentiments in several domains. In [8], authors proposed a word2vec+CNN architecture that uses word2vec to compute the vector representation of words as the input for the 7-layers CNN model. Authors in [9] also proposed deep learning models for sentiment analysis based on CNN architecture but with different convolutional filter sizes.

Some recent works propose an RNN-CNN architecture in the NLP context. Zhang et al. applied RNN to support the convolutional layers in order to capture long-term dependencies across the whole document [10]. Sosa in [11] presented a twitter sentiment analysis by merging CNN and LSTM models. In addition, Hassan et al. proposed a ConvLstm neural network architecture that employs both convolutional and recurrent layers on top of a pre-trained word vector [12]. This architecture is able to capture syntactic and semantic information among word representations. Also, the convolutional layer can extract high-level features from input sequence of sentences efficiently. In our paper, we provide a comparison between regular CNN, LSTM and the combined models, the results in the paper show that the LSTM-CNN model achieves the most performing results.

Several works apply the power of the CNN-LSTM combination in other languages. [13] presents a multichannel LSTM-CNN model for Vietnamese sentiment analysis where it captures both local and global dependencies in a sentence. Authors of [14] investigate the benefits of combining CNN and LSTM networks in an Arabic sentiment classification task by using character level to increase the number of features of each sentence.

Although sentiment analysis has been carried out in some domains such as tweets, movie reviews; there are limited contributions in the area of MOOCs due to the lack of MOOCs labeled datasets for training and evaluation. In [15], authors presented a new contribution in the field through a real case study where they examined learners' emotions and behavioral patterns in the MOOC. It also provided a comparison between different techniques, lexicon unsupervised approaches, and supervised machines learning approaches. Another work [4] evaluates the impact of sentiment on attrition over time by exploring the correlation between sentiment ratio and the number of students who dropout each day. In this work, our main contributions are to first explore the

power of Deep Learning, especially the combination CNN-LSTM, to measure the sentiment analysis in MOOC discussion boards and reviews. We then investigate the effects of quantifying uncertainties in sentiment analysis through the usage of a Bayesian Deep Learning model.

3 Background

3.1 Convolutional neural network CNN

A CNN is a special type of neural networks mainly used in the field of computer vision and NLP for classification or segmentation tasks. Convolution is the process of taking input data and selecting features matrix from it [16]. The embedding from different words of a sentence are stacked together to form a two-dimensional array, then the result is transferred to a convolution layer to produce a new feature representation. Then pooling methods are applied on new features to form the hidden representation. The final layers are fully connected in order to make the final prediction. We use a softmax function to get the final classification.

3.2 Long short-time memory (LSTM)

LSTM is a popular recurrent neural network architecture, which is a class of neural networks that has recurrent connections between units. It was first proposed by Hochreiter and Schmidhuber (1997) to capture long-term dependencies. LSTM uses linear memory cells surrounded by multiplicative gate units to store read, write and reset information [17].

LSTM has three gates: the input gate, the forget gate and the output gate. The LSTM gets the input from the current time-step and the output from the previous time-step, and produces an output, which is fed to the next time-step.

The LSTM networks are designed mainly for sequence prediction problems. They work by learning a function "f" that maps input value (X) onto output sequence Y.

$$Y(t)=f(X(t)) \tag{1}$$

3.3 Bayesian deep learning model

Although deep learning models have achieved state of the art results on various tasks, they do not provide the confidence of their classification or regression predictions. Recent proposals aim to make better decisions by re-casting Deep Learning models as Bayesian and hence introducing uncertainty to predictions. Bayesian Deep Learning [18] is a field at the intersection of Deep Learning and Bayesian probability theory. It provides principled uncertainty estimates to deep learning architectures. Suppose that our network is parameterized by a set of weights W for a dataset D. A Bayesian deep learning network aims to find the posterior distribution P(W/D) instead of a point estimate. The inference results are a predictive distribution:

$$P(D) = p(x,y) = \int p(y|x, w)p(w)dw \quad (2)$$

Where D can be written as x,y input, output pairs.

There are three principle Bayesian deep learning approaches. The table 1 gives a summary of each method [18]:

Table 1. comparative table of different Bayesian deep learning approaches

Approaches	Description	Advantages	Drawbacks
Approximation integrals with Markov Chain Monte Carlo MCMC	Generate samples from the posterior distribution by constructing a reversible Markov-chain that has as its equilibrium distribution the target posterior distribution	- Leads to great results	- Takes a long time to compute
Variational Inference	An approach to estimate a density function by choosing a Gaussian distribution and progressively changing its parameters until it looks like the one we want to compute, the posterior.	- Faster than MCMC Edward libraries	- Slow for very deep Bayesian network - Not very performant
Monte Carlo Dropout	A Bayesian interpretation of the regularization technique dropout using dropout for every layer during training as well as testing. This is the equivalent to sampling from a Bernoulli distribution and provides a measure of model's certainty	- Easy to turn an existing deep network to Bayesian network - Faster than the other technique - Does Not require an inference framework	- Expensive and requires computational resources in real time due to dropout activation in test time

In this paper, we implement Monte-Carlo dropout based on its advantage cited above and relevant results presented in several works [19], and it's a faster way to describe uncertainty over sentiment predictions. The dropout is used in deep learning models at the training time to avoid over-fitting. According to Gal [18] using dropout at inference time is equivalent to doing Bayesian approximation. In practice, in Monte Carlo dropout, the dropout is applied at both training and inference time. The prediction is not deterministic because the model could predict different values each time. The sample mean is the estimation and for the uncertainty, we simply calculate the sample variance as illustrated in the formulas below:

$$E(y*) \approx \frac{1}{T}\sum_{t=0}^{T} \hat{Y}_t^{*} (X^*) \quad (3)$$

$$Var(y^*) \approx \tau^{-1} + \frac{1}{T}\sum_{t=1}^{T} \hat{y}_t^*(x^*)^T \hat{y}_t^*(x^*) - E(y^*)^T E(y^*) \quad (4)$$

τ relates to the precision of the Gaussian process model, it's used in the calculation of the predictive variance.

$$T=\frac{l^2 p}{2N\lambda} \quad (5)$$

Where l is a user defined length scale, p is the probability of units not being dropped; N is the number of training samples and λ is a multiplier used in the regularization such as the weight decay.

4 Proposed architecture: Bayesian CNN-LSTM architecture for sentiment analysis

In this paper, we aim to determine the polarity of unstructured data written in natural language. Therefore, the pre-processing steps are needed in order to transform text data into a form that can be processed computationally. After the pre-processing steps, many deep learning algorithms are used to conduct the sentiment pattern recognition. Figure 1 presents our sentiment analysis pipeline.

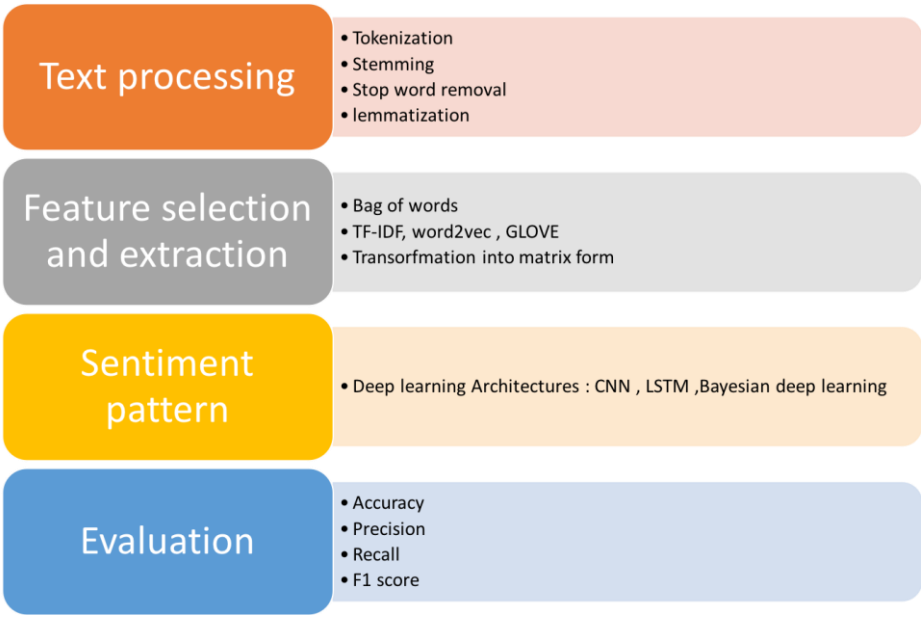

Fig. 1. The proposed sentiment analysis pipeline

4.1 Text processing

Text processing is used to extract interesting information from unstructured data. We performed data pre-processing steps on our dataset before feeding them into our deep learning model. The pre-processing is performed by various operations. Namely: Stemming, Stop word, Tokenization and Lemmatization

4.2 Feature selection and extraction: Word embedding

This is the second step in our sentiment analysis model. Its main objective is the selection of the best technique to extract features from the text, and these features can later be used in the deep learning model. We use a technique called word embedding to transform text into matrix form, because deep learning networks cannot process text input and perform operations. This layer associates each word in the input data to a vector representation to produce an embedding matrix.

Word embedding is a type of algorithms which map word from vocabulary to a real-valued vector by embedding semantic and syntactic meaning obtained from unlabeled corpus. It allows preparing the data in the appropriate vector format for the input of neural networks. It's a powerful technique widely used in natural language processing tasks such as semantic analysis, information retrieval, and machine translation.

Word embedding techniques are devised into traditional and distributional methods. The bag of words BOW vector representation is the most used traditional vector representation. To represent word vector, it counts how many times a word appears in a document. This approach assumes however that two sentences are similar if they use the same words.

The second traditional approach is based on TF_IDF weights. It firstly measures the term frequency TF which determines how important a word is by looking at how frequently it appears in the document.

$$\text{TF(t)}= \frac{numberoftimestermstappearinadocument}{totalnumberoftermsinthedocument}$$

Secondly, Inverse document frequency IDF is used to calculate the weight of rare words across all documents.

$$\text{IDF(t)}=Log(\frac{Totalnumberofdocuments}{Numberofdocumentswithtermtinit})$$

The TF_IDF value increases proportionally to the number of times a word appears in a document.

However, these traditional approaches are unable to capture words meaning. The distributional methods solve this shortcoming by using a contextual similarity. In this paper, two different algorithms are used. The first one is word2vec and the second one is GLOVE.

The word2vec algorithm was first introduced in 2013[20]. Word2vec utilizes Continuous Bag of Words (CBOW) or Skip-gram models for computing word vectors. The CBOW model learns the embedding by predicting the current word based on its context (i.e., surrounding words). The skip-gram model learns by predicting the surrounding words (context) given a current word.

Glove, on the other hand, was proposed after word2vec and introduced a few changes compared to the skip-gram model. The Glove algorithm considers global statistics (word co-occurrence, count based models) to obtain the word vectors while word2vec algorithm considers only the local statistics (local context information of words, windows-based models).

We aim to represent an input word Wi as one-hot vector based on a corpus of 5000 words. This vector will have 5000 components and we'll place 1 in the corresponding position to Wi and 0 in all the other positions. The output of the network is a single vector with 5000 components and contains for every word the probability that a randomly selected nearby word is that vocabulary word. We use a 300 features and SoftMax function in the output neurons. The figure 2 shows the architecture of our network.

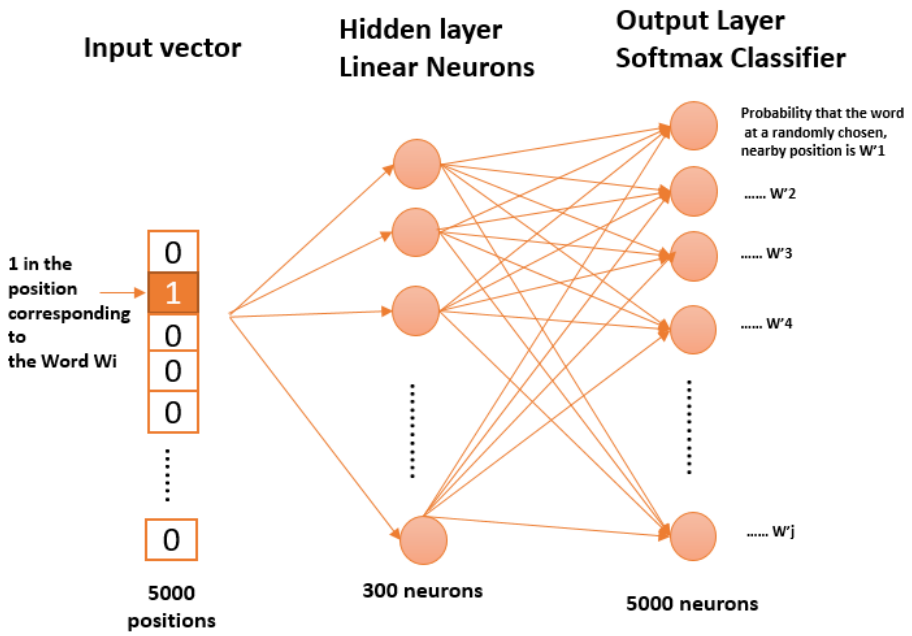

Fig. 2. skip-gram neural network

In our study, we used a 300-dimension word2vec embeddings trained on Google News with about 100 billion words and 300-dimension and a GloVe embeddings trained on twitter dataset with about 2 billion tweets and 27 billion tokens

4.3 Sentiment pattern recognition

This is the main step in the sentiment analysis process. The sentiment analysis can be applied on three levels of granularity, word level which is the analysis that determines the polarity of a word, sentence level that determines the polarity of a sentence and it's often used in social networks. Finally, the most difficult level is the document level which determines the polarity of a document. In our work, we use a sentences level granularity, because it is specific to social network and largely used in the analysis of opinion.

Uncertainty allows to indicate if a model is under-confident or falsely over-confident, and hence can help get better performance out of a model. In our work, we study the effects of quantifying uncertainty model and input-dependent data uncertainty in sentiment analysis. Bayesian deep learning is one approach to quantify uncertainty associated with model parameters. It represents all model weights as probability distributions over possible values instead of fixed scalars. We adopt an architecture that combines a CNN with LSTM networks in order to capture deep sentiment features. According to several works, combining the features of each distinct model can result in a complete and powerful model that improves the prediction performance [17]. In our case, CNN captures the local information and LSTM captures long dependencies. Our model is composed of an embedding layer, a convolutional layer, a max-pooling layer, an LSTM layer and a fully connected layer. In the Bayesian CNN-LSTM, we add a dropout layer with a constant dropout rate P before each layer. The output of our Bayesian model is a probability distribution and not a point estimate as presented in the figure 3.

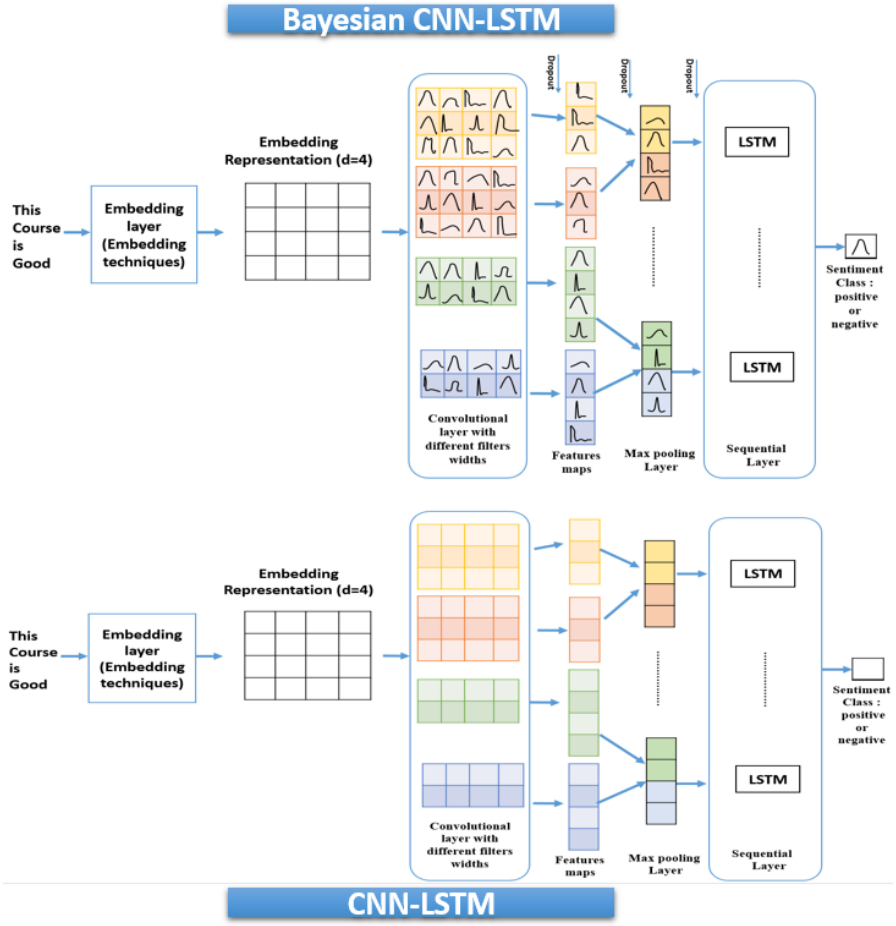

Fig. 3. CNN –LSTM vs Bayesian CNN-LSTM Architectures

The Bayesian CNN-LSTM model consists of an initial input layer of the network which is the MOOCs Reviews. These reviews are tokenized into words and each word is mapped to a word vector representation (word embedding), such that an entire text can be mapped to a matrix of size n x d, where n is the number of the words in the text and d is the dimension of the embedding space. Then, the multiple convolutional filters slide over the matrix to produce a new feature map and the filters have different sizes to detect and generate specific features or patterns. These filters are working in parallel to generate multiple features maps.

The filter slides from left to right starting in the top left corner and finishing in the bottom right corner jumping down one row at a time. After every training, session the weights in the filter are updated using backpropagation. Then, the max-pooling layer calculates the maximum value as a corresponding feature to a specific filter. The reason of applying this max operation to the result of each filter is to eliminate non-maximal values and thus reducing the computation for upper layers, and to extract the local dependency within different regions to keep the most salient information.

The outputs of the Max-pooling layer become inputs to the LSTM networks to measure the long-term dependencies of feature sequence. One of the advantages of the LSTM is the possibility to capture long-distance dependency across regions by considering the previous data. This sequential layer integrates each region vector into a text vector. In our case LSTM could allow us to capture changing sentiment in a learner's review or forum posts.

The outputs of the LSTMs are concatenated and fed to a fully connected layer, and an activation function SoftMax is applied to generate the final output prediction i.e., positive or negative text. The advantage of this model is the initial convolution layer will extract local features and the LSTM will then be able to use ordering of said features to learn about the inputs text ordering.

For implementation of Bayesian deep learning model, the Monte-Carlo dropout (MC-dropout) approximates the posterior distribution by a product of the Bernoulli distribution, the so-called dropout variational distribution. In practice, optimizing Bayesian deep learning with the MC-dropout is technically equivalent to the standard deep learning with dropout as regularization. In contrast to standard deep learning that predict outputs by turning-off the dropout at the inference phase, the MC-dropout networks keep turning on the dropout and predict outputs by sampling and averaging them, which theoretically corresponds to integrating the posterior distribution and likelihood.

5 Experimentation and analysis

The use of deep learning supervised approach in the sentiment analysis in MOOC has several limitations mainly the need of dataset for training. MOOC platforms provide only raw data for reviews or forum posts, this data needs to be labeled manually which is considered as a very time consuming task. However, the labeling process is very subjective and it needs to be treated by more than one person to ensure data validity. Therefore in our work we based on 100K Coursera's Course Reviews to evaluate the performance of our Bayesian CNN-LSTM model. Then the pre-trained model is

used to measure the sentiment analysis in MOOC forums in order to evaluate the correlation between sentiment analysis and MOOC attrition rate.

5.1 Evaluation of Bayesian CNN-LSTM sentiment analysis model

Dataset. The experimentation and analyses in this paper are based on 100K Coursera's Course Reviews Dataset available on https://www.kaggle.com/septa97/100k-courseras-course-reviews-dataset. Each entry in this dataset represents a single review for a particular course. The data was collected from the website of Coursera and pre-labeled depending on their rating. For 5 rating, the review was labeled as very positive, positive for 4, Neutral for 3, Negative for 2, and very Negative for 1. The Dataset contains 1835 different courses and 140320 reviews. Figure 4 shows the most popular courses according to the reviews number. We notice that Machine Learning by Andrew NG has the most ratings.

Fig. 4. Top 20 Courses in Coursera Platforms

Experimental setup. Our goals in this experimentation are to determine the best deep learning method for analyzing sentiment of text and quantifying uncertainty in sentiment analysis prediction. In this paper, we built different CNN and LSTM models using several hyper-parameters for each layer. The best parameters for our models were selected based on a grid search on 35% of the dataset. The used parameters are presented in the table 2.

In the experiments, the dataset was split into training, validation and testing sets following the percentages 70%, 10%, and 20% respectively. We select 10% of the training data as a validation set. We measure the validation loss in each training epoch through the validation loss and the training will stop when we observe at least two consecutive epochs with no improvements. In addition, the test set is used to measure the performance of the model. The model was implemented using the Tensorflow library.

Table 2. Model hyper parameters

Hyper-parameters	Values
Filter size	[3,5,7]
Number of filters	150
Activation function	ReLU
Embed size	400
LSTM hidden dimension	300
Number of epochs	50
Learning rate	0.001
Batch size	64
Dropout rate	0.2
Regularizer	L2

Results & Discussion. The performance of our model is evaluated using standard evaluation metrics, we measured the precision, recall, F1-measure, and accuracy as shown in equations 6,7,8,9.

$$\text{Precision (P)} = \frac{TruePositive}{(TruePositive + FalsePositive)} \tag{6}$$

$$\text{Recall (R)} = \frac{TruePositive}{(TruePositive + FalseNegative)} \tag{7}$$

$$\text{F1-Score (F1)} = \frac{2*(Precision*Recall)}{(Precision + Recall)} \tag{8}$$

$$\text{Accuracy} = \frac{TruePositve + TrueNegative}{Total} \tag{9}$$

We implemented two common deep learning models CNN, LSTM, and proposed hybrid Bayesian CNN-LSTM on the Coursera reviews dataset. We used word2vec and glove to learn words' vector representation. The results of the proposed models are shown in table 3.

The CNN-LSTM model achieves a satisfactory accuracy and F1-score of 90.01% and 88%, 1.09% higher than CNN and 2.47 % higher than LSTM. The results show that the CNN is able to recognize the local patterns and the LSTM takes into account the long-term dependency of the text and is able to harness the text ordering. Furthermore, combining the advantages of CNN and LSTM gives an optimistic result.

We evaluate the non-Bayesian CNN-LSTM against the Bayesian CNN-LSTM. The first model output was a single point estimate with dropout only in the training step. The Bayesian model includes dropout in the training and testing step. According to the experimental results shown in Table 3, we can see that the Bayesian CNN-LSTM model outperform the non-Bayesian model. It boosts performances by 1.26%.

Table 3. Comparison of proposed Bayesian CNN-LSTM model with CNN, LSTM, CNN-LSTM, precision , Recall , F1-score and accuracy

	Coursera MOOCs review			
	CNN	***LSTM***	***CNN-LSTM Non-Bayesian***	***CNN-LSTM Bayesian***
Precision	0.87	0.85	0.89	0.90
Recall	0.84	0.83	0.86	0.85
F1-score	0.86	0.84	0.88	0.88
Accuracy (%)	88.92	87.54	90.01	91.27

5.2 Evaluation of the correlation between sentiment analysis

One of the major issues in MOOCs is the high attrition rate of students. Several researchers propose to reduce the dropout rate through the prediction of the students at risk of dropping out during courses then they recommend them a suitable pedagogical strategy or complementary resources to help them complete their courses.

To build a dropout predictor in MOOCs, we need to identify several features to have a better prediction performance. We aim in this part to show the importance of considering the sentiment analysis as one of the most relevant features in the dropout prediction model. The goal of this experimentation is to investigate the link between sentiment analysis of students' forum posts and the dropout in MOOCs. We use our pretrained Bayesian CNN-LSTM sentiment analysis model and test it on MOOC Coursera review dataset to detect learners at risk of dropping out.

The experimentation and analyses in this part are based on a dataset which was prepared by Stanford's university exclusively for this work1. The data were collected from a MOOC introduction to computer science which was launched in March, 2016. The course lasted twelve weeks with 11 607 participants at the beginning of the week and 3 861 participants staying until the last week of course. Globally, 20 828 learners participated, with a dropout rate of approximately 81,4 %. Figure 5 shows that the probability of dropout is at its highest during the first two weeks.

Based on our Bayesian CNN-LSTM sentiment analysis model, we measure the polarity of the students' forum posts in every week. To evaluate the correlation between the sentiment analysis and attrition rate, we analyze the correlation between the number of students who dropout from the courses every week and the sentiment ratio expressed in the discussion forums every week as illustrated in figure 6. We measure the correlation coefficient between this two parameters R= -0.8594 and we notice a strong negative correlation because as the number of students who quickly drop out of the course decreases, the sentiment ratio increases.

[1] http://datastage.stanford.edu/

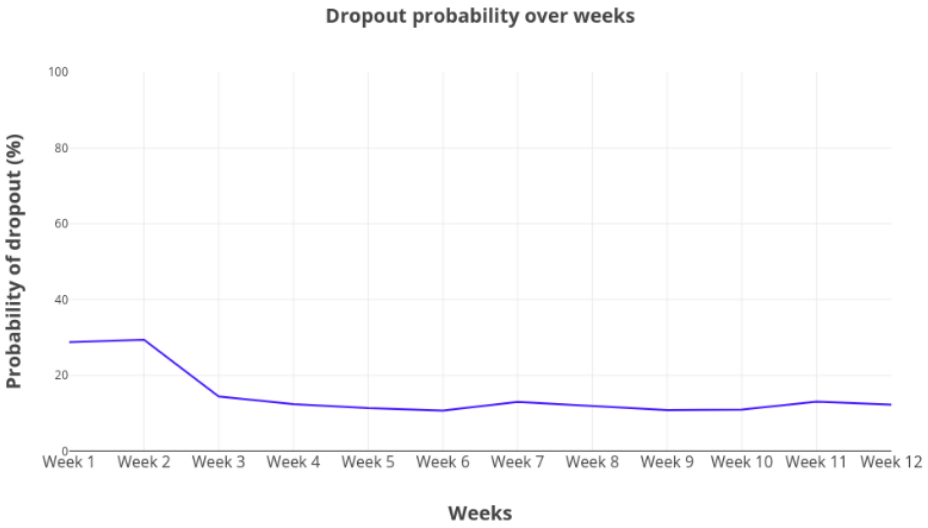

Fig. 5. Dropout probability over time

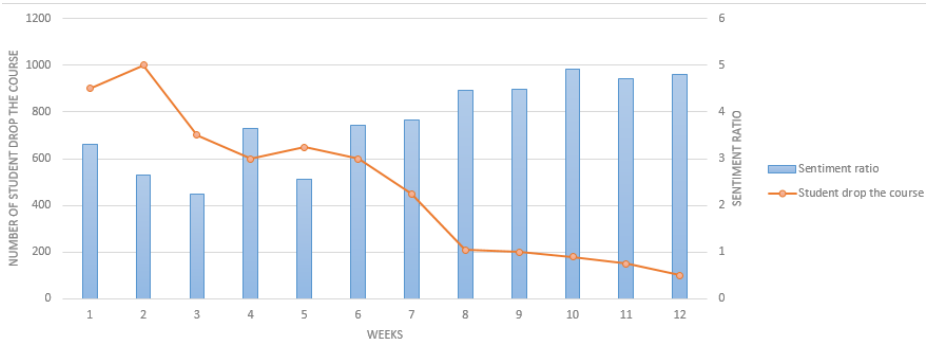

Fig. 6. The Proportion of Sentiment ratio and active students over time

Based on our sentiment analysis model, we measure the average sentiment of all the posts in discussion forum for 10 students enrolled in this course and analyze the correlation with the dropout. When the dropout label of the learner is 0, this indicates that the learner will continue to learn. Otherwise, the dropout label is 1 which means the learner will quit the course.

Figure 7 shows the proportion of the average sentiment of every student and her dropout rate. We measure the correlation coefficient R=0.8128 and notice a strong positive correlation. When the average sentiment increases (between 4 and 5), the student completes the course. In contrast, when the average of the sentiment decreases (between 1 and 3), the student abandons the course.

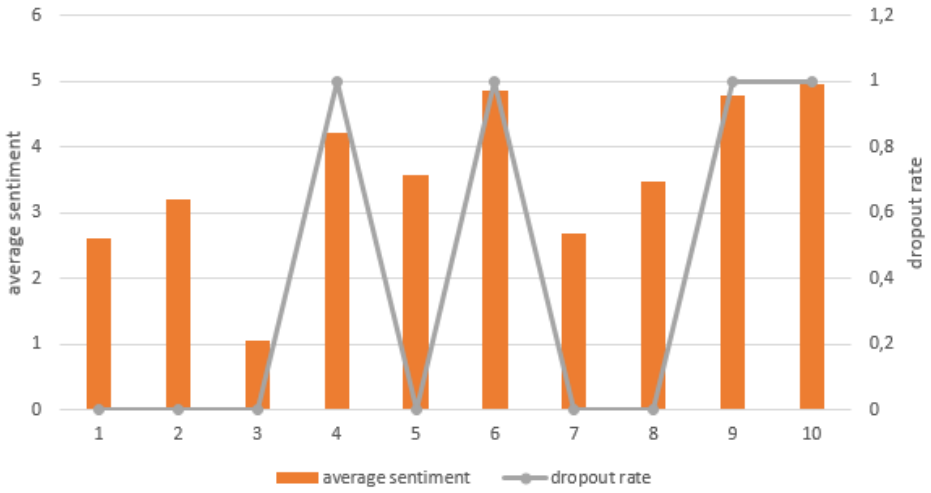

Fig. 7. Proportion of the average sentiment and dropout rate

6 Conclusion

MOOCs have become more and more popular because of inherent characteristics that distinguish them from other online education courses (completely open, flexible in access, and free to anyone). However, only a limited number of enrolled learners complete the courses due to several factors: learners-related such as learners' motivation and MOOC-related such as unsuitable courses design.

In an attempt to reduce the dropout rate, we propose to analyze sentiments expressed by learners in forum posts within a MOOC. This will allow us to, firstly, measure learners' satisfaction and the success of the MOOC which can help improve the quality of the course. Secondly, it aims to classify the learners in different groups to ensure the personalization and adaptation of the courses. Finally, it allows building a predictor based on sentiment analysis to predict learners at risk of dropping out during the course.

The task is challenging because forum posts are written in human language, which is incredibly complex. Teaching a machine to analyze the grammatical nuances, sentiments, and meaning is thereby a difficult process. To address these issues, we propose a Bayesian CNN-LSTM model to measure the sentiment analysis in MOOCs platforms. Our model combines the deep learning architectures CNN and LSTM to achieve better performance on sentiment analysis tasks in MOOCs platforms. We then evaluated the benefits of quantifying uncertainties in modern deep learning models applied in the context of sentiment analysis.

We showed that the CNN-LSTM models render a better performance compared to the individual LSTM and CNN. The combination CNN and LSTM architecture captures both the local and global dependencies in a sentence. In addition, by quantifying uncertainties, model performances are improved on the sentiment analysis task. The Bayesian CNN-LSTM model achieves 91 % accuracy which is the best one compared to other deep learning architectures.

Furthermore, we investigated the link between sentiment analysis in students' forum posts and the dropout in MOOCs by using our Bayesian CNN-LSTM model. We noticed a strong positive correlation between the average sentiment of a student and her dropout rate (0.8128), and a strong negative correlation between the sentiment ratio and the number of active users every week (-0.8594).

As a future work, we aim to combine a bag-of-word model and embeddings to produce better pre-training features as inputs of the neural network. We aim to add more LSTM layers and bidirectional LSTM in the hybrid CNN-LSTM model to enhance performance. We also aim to validate the model over larger discussion forums' datasets. It is also important to apply this model on other NLP applications (text classification, personality identification).

7 References

[1] Brahimi, T. and Sarirete, A. (2015). Learning outside the classroom through MOOCs. Computers in Human Behavior, 51, pp.604-609. https://doi.org/10.1016/j.chb.2015.03.013

[2] Shah, D. Monetization Over Massiveness: Breaking Down MOOCs by the Numbers in 2016. EdSurge. Available online: https://www.edsurge.com/ (accessed on 25 July 2017).

[3] M. Laal and S. M. Ghodsi. Benefits of collaborative learning. Procedia - Social and Behavioral Sciences, 31(0):486 – 490, 2012. World Conference on Learning, Teaching and Administration - 2011. https://doi.org/10.1016/j.sbspro.2011.12.091

[4] Wen, M., Yang, D., & Rosé, C.P. (2014). Sentiment Analysis in MOOC Discussion Forums: What does it tell us? EDM.

[5] Omar Yousef Adwan, Marwan Al-Tawil, Ammar Huneiti, Rawan Shahin, Abeer Abu Zayed, Razan Al-Dibsi: Twitter Sentiment Analysis Approaches: A Survey. iJET 15(15): 79-93 (2020). https://doi.org/10.3991/ijet.v15i15.14467

[6] Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; and Wierstra, D. 2015. Weight uncertainty in neural networks. arXiv preprint arXiv:1505.05424.

[7] Bahdanau et al., 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv: 1409.0473, 2014.

[8] X. Ouyang, P. Zhou, C. H. Li, and L. Liu, "Sentiment analysis using Convolutional neural network," 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, Oct. 2015. https://doi.org/10.1109/cit/iucc/dasc/picom.2015.349

[9] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In EMNLP 2014 - Empiri- cal Methods in Natural Language Processing, pages 1746–1751, August. https://doi.org/10.3115/v1/d14-1181

[10] Zhang, J., Li, Y., Tian, J., & Li, T. (2018). LSTM-CNN Hybrid Model for Text Classification. 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 1675-1680. https://doi.org/10.1109/iaeac.2018.8577620

[11] Sosa, Pedro M. "Twitter Sentiment Analysis Using Combined LSTM-CNN Models". Konukoii.Com, 2018, http://konukoii.com/blog/2018/02/19/twitter-sentiment-analysis-using-combined-lstm-cnn-models/. Accessed 22 Feb 2018. https://doi.org/10.1109/ccis.2018.8691381

[12] Hassan, A., & Mahmood, A. (2017). Deep Learning approach for sentiment analysis of short texts. 2017 3rd International Conference on Control, Automation and Robotics (ICCAR). https://doi.org/10.1109/iccar.2017.7942788
[13] Vo, Q., Nguyen, H., Le, H.B., & Nguyen, M. (2017). Multi-channel LSTM-CNN model for Vietnamese sentiment analysis. 2017 9th International Conference on Knowledge and Systems Engineering (KSE), 24-29. https://doi.org/10.1109/kse.2017.8119429
[14] Alayba A.M., Palade V., England M., Iqbal R. (2018) A Combined CNN and LSTM Model for Arabic Sentiment Analysis. In: Holzinger A., Kieseberg P., Tjoa A., Weippl E. (eds) Machine Learning and Knowledge Extraction. CD-MAKE 2018. Lecture Notes in Computer Science, vol 11015. Springer, Cham. https://doi.org/10.1007/978-3-319-99740-7_12
[15] Moreno-Marcos, P.M., Alario-Hoyos, C., Merino, P.J., Estévez-Ayres, I., & Kloos, C.D. (2018). Sentiment analysis in MOOCs: A case study. 2018 IEEE Global Engineering Education Conference (EDUCON), 1489-1496. https://doi.org/10.1109/educon.2018.8363409
[16] Yamashita, R., Nishio, M., Do, R.K.G. et al. Convolutional neural networks: an overview and application in radiology. Insights Imaging 9, 611–629 (2018). https://doi.org/10.1007/s13244-018-0639-9
[17] M. Sundermeyer, R. Schluter, and H. Ney, "LSTM neural networks for language modeling.," in Interspeech , pp. 194–197, 2012. https://doi.org/10.21437/interspeech.2012-65
[18] Y. Gal, Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning", International Conference on Machine Learning (ICML 2016), pp. 1050-1059, June 2016.
[19] Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; and Wierstra, D. 2015. Weight uncertainty in neural networks. arXiv preprint arXiv:1505.05424
[20] : T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems. 2013, 3111–3119.

8 Authors

Khaoula Mrhar, is with IPSS Research Team, FSR, Mohammed V University, Rabat, Morocco.

Lamia Benhiba, is with ENSIAS, Mohammed V University, Rabat, Morocco.

Samir Bourekkache, is with Smart computer science laboratory (LINFI), University of Biskra, Computer science department, Biskra, Algeria.

Mounia Abik, is with Information Retrieval and Data Analytic (IRDA) Team, ENSIAS, Mohammed V University, Rabat, Morocco

Article submitted 2021-06-01. Resubmitted 2021-10-01. Final acceptance 2021-10-01. Final version published as submitted by the authors.