Bidirectional CRNN and BERT Embeddings for Sentiment Analysis on Twitter Data

Simone Libutti, Elisa Martinez, Anisha Mohamed Sahabdeen, Lara Nonino
Team name: Just CILlin
Department of Computer Science, ETH Zurich, Switzerland

***Abstract*—This project aims to perform sentiment analysis on a dataset consisting of tweets. We start by considering two baseline results obtained through simple machine learning techniques with different embeddings and compare these to a more complex model centered around a state-of-the-art architecture for this kind of task, using BERT embeddings. In the comparative study, we evaluate the classification accuracy of each of the models.**

I. Introduction

The task of sentiment analysis in machine learning is a classification problem that aims to determine whether a given text can be associated with a positive or a negative feeling. This type of analysis is usually customer reviews related to various services (e.g., restaurants, movies, online shopping) or social network posts. The instance problem we tackle in this work falls in the latter of these two categories: more specifically, we are concerned with sentiment analysis of tweets.

As with most modern approaches to natural language processing tasks, the available literature mainly revolves around Transformer architectures, and there is where we have centered our efforts.

The approach we followed starts with preprocessing operations on the tweets, which will be detailed in the upcoming section. Following this step, we analyzed the dataset using two baseline models. Our proposed baselines consist of two simpler machine learning models: a 1-dimensional convolutional neural network and a bidirectional recurrent neural network. Each of these architectures has been tested with three different word embeddings, and we have collected the results for a more straightforward comparison with our model. Further details on these architectures and the embeddings used are described in Section II.

Following the baseline tests, we proceeded with experimentation on what we refer to as our main model, developed through a combination of the baseline systems and making use of a pre-trained BERT model to produce more meaningful embeddings. This led to a richer architecture and a noticeable increase in performance when compared to the plain 1D-CNN and BiRNN approaches.

II. Methods

A. Word2Vec and GloVe embeddings

Two widely employed word-embedding models used for text transformation are Word2Vec [1] and GloVe [2].

Word2Vec algorithms utilize a neural network model based either on Skip-gram [3] or CBOW [4]. In both cases each unique word is represented by a list of vectors. The cosine similarity between these vectors serves as the mathematical function for selecting the appropriate vector, indicating the degree of semantic similarity between the words.

Unlike Word2Vec, GloVe takes a different approach to generating word embeddings. It focuses on the co-occurrence statistics of words in a large corpus of text to learn word representations. The core idea is to build a global word-word co-occurrence matrix that captures how frequently words appear together in the same context within a window of words. GloVe then uses this co-occurrence matrix to factorize the word vectors in a way that the resulting word embeddings preserve the global semantic relationships between words. This means that it not only captures local context information but also considers the overall word co-occurrence patterns in the entire corpus.

In our case, we investigated the performance of both pre-trained Word2Vec and GloVe embeddings.

B. 1-Dimensional CNNs

Convolutional Neural Networks have been employed for NLP in the past [5]. 1-D CNNs in particular are a variant of traditional Convolutional Neural Networks specially designed to handle sequential data. Unlike standard CNNs, which primarily process images with two-dimensional grids of pixels, 1-D CNNs work on one-dimensional sequences of data, perfectly serving our purposes in which data consists of short text sequences.

The fundamental building block of a 1-D CNN is the convolutional layer. During the training process, the 1-D CNN learns to optimize the filter weights to detect relevant features from the input sequence. The typical architecture of a 1-D CNN consists of a sequence of convolutional and pooling layers, followed by one or more fully connected layers for classification or regression tasks.

The advantages of using CNNs for sequential data lie in their ability to automatically learn relevant patterns and features from the data, reducing the need for manual feature engineering. Furthermore, they exhibit translation invariance, meaning they can identify patterns regardless of their exact location in the sequence, making them robust to slight variations in input data.

C. Bidirectional RNNs

Bidirectional Recurrent Neural Networks [6] are specialized RNNs that process sequential data bidirectionally, considering both past and future inputs. By incorporating two separate recurrent layers—one processing data forward and the other backward—Bidirectional RNNs capture comprehensive context from the entire sequence.

Training involves optimizing the network's weights to learn complex temporal patterns, making them valuable in natural language processing. In our case, we have combined them with an Bi-LSTM module to handle long-range dependencies and vanishing gradient issues.

The typical architecture includes forward and backward recurrent layers, followed by additional layers for classification, sequence labeling, or machine translation tasks.

D. BERT and Transformers

Differently from convolutional and recurrent architectures, Transformers [7] fully rely on self-attention. This allows the model to weigh the importance of different words (or tokens) in a sentence with respect to each other. The model assigns attention scores to each word in the input, indicating how much attention it should pay to other words during processing. This mechanism allows the model to capture dependencies between words regardless of their positions in the input sequence.

To capture different types of relationships between words, the transformer uses multiple attention heads in the self-attention mechanism. Each head focuses on different aspects of the input sequence and learns different representations. This enables the model to learn diverse features from the input data.

BERT (Bidirectional Encoder Representations from Transformers) [8] allows for bidirectional context-sensitive representations of words leveraging the power of this architecture, all while remaining task-agnostic. This allows for a noticeable increase in performance when compared to the embeddings we have used in the baselines, namely Word2Vec and GloVe, as these are context-independent and don't take into consideration the surroundings of each word in the sentences.

In our scenario, we employ the DistilBERT model [9], which is trained using knowledge distillation from the BERT architecture. It incorporates 40% fewer parameters while preserving 95% of BERT's performance. Consequently, the DistilBERT model is more lightweight and operates 60% faster than BERT, making it a highly efficient option for our specific task.

III. Baselines

We have constructed two baseline models to which we compare the final architecture.

Our initial baseline first applies three convolutional layers each with 100 channels and respective kernel sizes of 3, 4, and 5. For each convolutional output an adaptive average pooling layer is applied returning a vector of size 100. We then concatenate the three outputs producing a vector of size 300. Finally, the model concludes with a fully connected layer serving as the classification layer.

In our second baseline implementation, the entire sequence is encoded by a bidirectional RNN. More concretely, the hidden states of the bidirectional LSTM at both the initial and final time steps are concatenated as the representation of the text sequence. This single text representation is then transformed into output categories by a fully connected layer with two outputs, as in the CNN case.

We conducted various experiments comparing these baselines using different embeddings, as described in Section V.

IV. Our Model

Our architecture was inspired by the model presented by Kokab et al. in [10]. We combined the different methods presented in Section II as shown in the diagram in Figure 1.

Our architecture consists of three distinct blocks. Initially, we employ the pre-trained BERT model DistilBERT to extract sentence-level semantics and contextual features from the input data, generating embeddings. This allows for deeper precision compared to the simpler embeddings used in the baselines.

Then, the 768-dimensional embeddings are fed into a 1D-CNN with three layers. Compared to the original paper that uses dilated CNNs to increase the receptive field, we use 1D-CNNs. Their nature works well for the processing of tweets, which typically have a maximum length of 140 characters. In the CNN, we concatenate the output of 3 convolutional layers with kernel sizes of 3, 4, and 5, as in the baselines. The output is a sequence of vectors of size 300 (one for each token in the sentence) which is passed through a dropout layer with a dropout rate of 0.5. After that, the output is further processed by a linear layer, resulting in another sequence of vectors of size 100.

As a last step, we feed the output of the 1-D CNN into a bidirectional LSTM. The Bi-LSTM consists of 2 layers, each with 100 hidden states. The output is then forwarded through a linear layer. Next, a dropout layer with a rate 0.3 is used to introduce regularization and prevent overfitting. Finally, the vector is passed through another linear layer that transforms it into a binary output, classifying the sentiment as either negative or positive.

V. Experiments

A. Baseline experiments

To analyze the performance of the baselines, we used different embedding types: pre-trained Word2Vec from the Google-News-300 dataset and pre-trained GloVe from the Wiki-gigaword dataset with vector sizes 200 and 300. We

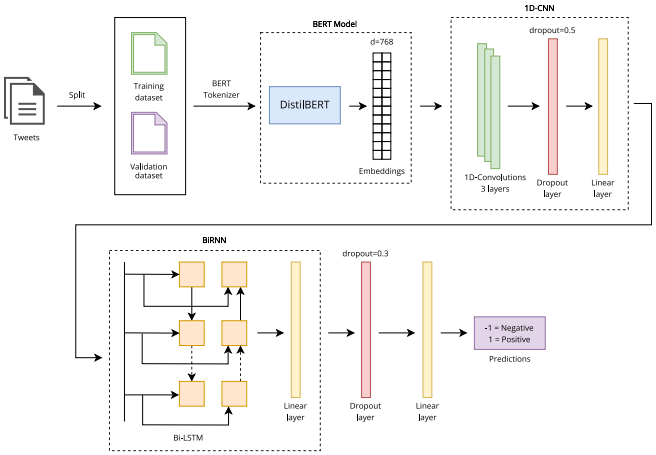


Figure 1. Model architecture

Model parameters	
DistilBertModel	66.4 M
CNN1d	952 K
BiRNN	443 K
Linear	202
Total	**67.8 M**

Table I
MODEL PARAMETERS

have collected experimental data regarding the performance of the baseline models in Table II.

These results were obtained after 5 epochs of training for the CNN and 6 epochs of training for the RNN with a learning rate of 10^{-3} and a batch size of 256.

Baseline Performances			
Embedding	Model	Size (number of parameters)	Validation accuracy
GloVe-100	CNN	120K	0.812034
GloVe-200	CNN	240K	0.829137
Word2Vec-300	CNN	360K	0.808812
GloVe-100	BiRNN	404K	0.870485
GloVe-200	**BiRNN**	**484K**	**0.874202**
Word2Vec-300	BiRNN	564K	0.856161

Table II
BASELINES RESULTS

As we can observe in Table II, the model that performs best is the bidirectional LSTM that uses GloVe embeddings with 200 features. We can conclude that using more features with GloVe improves the performance and that bidirectional LSTMs are better than 1D-CNNs since they cover the whole context of the sentence.

When comparing the baseline with the best performance to our model, we can observe an improvement in validation accuracy as seen in Table III.

B. Preprocessing experiments

In this section we present different experiments using data preprocessing. We analyzed various preprocessing steps suitable for the task. The training dataset consists of 1.25 M tweets labeled as positive and 1.25 M as negative. The dataset had already been preprocessed by (i) replacing user references with a generic token (`<user>`), (ii) replacing URLs with a generic token (`<url>`), (iii) lowering the text, (iv) adding whitespaces around punctuation and character emojis.

We investigated the following normalization strategies:

- **Remove numbers.** Numbers do not usually add sentiment and can thus be stripped away.
- **Remove stop words.** Stop words are extremely common words which are not useful for determining context or adding sentiment to a sentence.

In the following graph we are comparing the use the preprocessing methods (filtered numbers and stop words) with raw data. These experiments were done over 3 epochs using the reduced dataset with a batch size of 16. As it is seen in Figure 2 the best performance is achieved by the model trained on raw data. This might be because the pertained transformer is able to gain more context from unfiltered data and produce more informative embeddings.

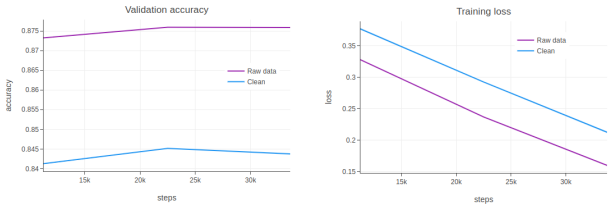


Figure 2. Data preprocessing experiments

C. Model experiments

When we ran our model, we noticed a trend where the training loss decreased, but the validation loss increased (Figure 3). This phenomenon is commonly associated with overfitting in Transformer-based models. In order to prevent this behaviour, we tried introducing a learning rate scheduler.

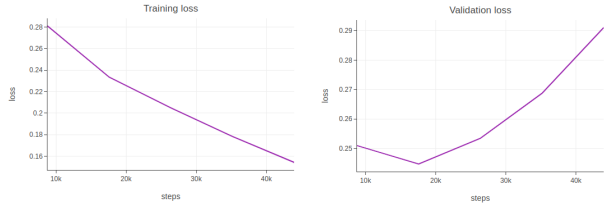


Figure 3. Validation and training loss of our main model

In Figure 5, we present a comparative analysis of various learning rate schedulers, all of which were implemented

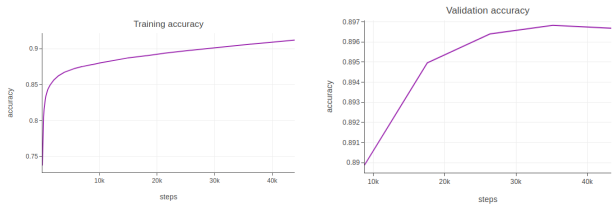


Figure 4. Validation and training accuracy of our main model

with an initial learning rate of 2e-5. The results, as depicted in the graph, clearly indicate that the model without any learning rate scheduling demonstrated superior performance compared to the other configurations. Therefore, in order to reduce the effect of overfitting, we interrupted the training at an early stage.

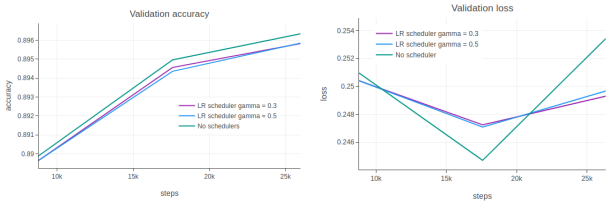


Figure 5. Validation accuracy and loss with schedulers

For our model selection, we opted to use the checkpoint from epoch 2 during training on raw data, without incorporating any schedulers. This particular epoch demonstrated the smallest loss value, as shown in Figure 4. As shown in Table III, our model outperforms the baseline, achieving the best performance results.

Model	Validation accuracy
BiRNN GloVe-200	0.874202
Our model	**0.895**

Table III
BASELINE VS OUR MODEL

VI. CONCLUSION

In conclusion, our research introduces a powerful combination of a BERT-based feature extraction approach with a 1-dimensional convolutional Bi-LSTM model for natural language processing tasks. Our final submission achieves an accuracy of 0.8974 as its public score on the provided dataset. The model demonstrates impressive results without relying on ensemble methods or excessive parameters. Its simplicity and efficiency make it a promising and practical solution for real-world applications.

REFERENCES

[1] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," 2013.

[2] J. Pennington, R. Socher, and C. Manning, "Glove: global vectors for word representation," *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014.

[3] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," 2013.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[5] Y. Kim, "Convolutional neural networks for sentence classification," 2014.

[6] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, 1997.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[9] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter," *CoRR*, vol. abs/1910.01108, 2019. [Online]. Available: http://arxiv.org/abs/1910.01108

[10] S. T. Kokab, S. Asghar, and S. Naz, "Transformer-based deep learning models for the sentiment analysis of social media data," *Array*, 2022.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Bidirectional CRNN and BERT Embeddings for Sentiment Analysis on Twitter Data

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Libutti	Simone
Martinez	Elisa
Mohamed Sahabdeen	Anisha
Nonino	Lara

With my signature I confirm that

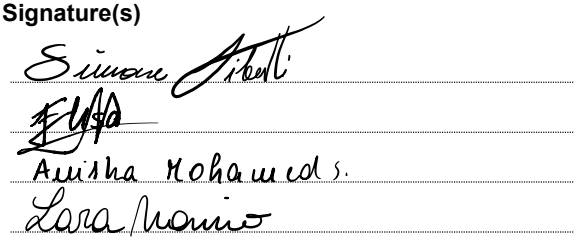
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date

Zürich, 31.07.2023

Signature(s)



For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.