Colón-Ruiz, C., Segura-Bedmar, I. (2020). Comparing deep learning architectures for sentiment analysis on drug reviews. *Journal of Biomedical Informatics*, 110

DOI: https://doi.org/10.1016/j.jbi.2020.103539

Comparing Deep Learning architectures for Sentiment Analysis on Drug Reviews

Cristóbal Colón-Ruiz*[a], Isabel Segura-Bedmar[a]

[a] *Computer Science Department, University Carlos III of Madrid, Avenida de la Universidad 30, 28911, Leganés, Madrid, Spain*

Abstract

Since the turn of the century, as millions of user's opinions are available on the web, sentiment analysis has become one of the most fruitful research fields in Natural Language Processing (NLP). Research on sentiment analysis has covered a wide range of domains such as economy, polity, and medicine, among others. In the pharmaceutical field, automatic analysis of online user reviews allows for the analysis of large amounts of user's opinions and to obtain relevant information about the effectiveness and side effects of drugs, which could be used to improve pharmacovigilance systems. Throughout the years, approaches for sentiment analysis have progressed from simple rules to advanced machine learning techniques such as deep learning, which has become an emerging technology in many NLP tasks. Sentiment analysis is not oblivious to this success, and several systems based on deep learning have recently demonstrated their superiority over former methods, achieving state-of-the-art results on standard sentiment analysis datasets. However, prior work shows that very few attempts have been made to apply deep learning to sentiment analysis of drug reviews. We present a benchmark comparison of various deep learning architectures such as Convolutional Neural Networks (CNN) and Long short-term memory (LSTM) recurrent neural networks. We propose several combinations of these models and also study the effect of different pre-trained word embedding models. As

*Corresponding author
Email address: ccolon@inf.uc3m.es (Cristóbal Colón-Ruiz*)

transformers have revolutionized the NLP field achieving state-of-art results for many NLP tasks, we also explore Bidirectional Encoder Representations from Transformers (BERT) with a Bi-LSTM for the sentiment analysis of drug reviews. Our experiments show that the usage of BERT obtains the best results, but with a very high training time. On the other hand, CNN achieves acceptable results while requiring less training time.

Keywords: Sentiment Analysis, Multi-Class Text Classification, Deep Learning, Convolutional Neural Network, Long short-term memory, Bidirectional Encoder Representations from Transformers

1. Introduction

Since the turn of the century, as millions of users' opinions are available on the web, sentiment analysis has become one of the most fruitful research fields in Natural Language Processing (NLP). Research on sentiment analysis has covered a wide range of domains such as economy, polity and medicine, among others. In the pharmaceutical field, the automatic analysis of online user reviews allows us to analyze large amounts of users' opinions and obtain relevant information about the effectiveness of drugs and their side effects, which could be used to improve the pharmacovigilance systems.

Throughout the years, approaches for sentiment analysis have progressed from simple rules to advanced machine learning techniques such as deep learning, which has become an emerging technology in many NLP tasks. Sentiment analysis is not oblivious to this success, and several systems based on deep learning have recently shown their superiority to former methods, achieving state-of-the-art results [1, 2, 3, 4] on standard sentiment analysis datasets. Thus, deep learning has been used extensively in sentiment analysis for may domains. However, very few attempts have been made to apply deep learning to sentiment analysis of drug reviews [5].

While sentiment analysis at document level (or sentence level) can be viewed as a text classification task where the goal is to assign a polarity to each text.

Most previous studies only deal with two or three polarities (positive, negative and neutral). A finer-grained polarity classification is more challenging due to the larger number of classes. To the best of our knowledge, this is the first attempt to conduct a finer-grained polarity classification of drug reviews.

In this work, we compare different deep learning such as Convolutional Neural Network (CNN), Long Short Term Memory (LSTM) and Bidirectional Encoder Representations from Transformers (BERT), three state-of-the-art deep learning models that have been used in different NLP tasks, including sentiment analysis. Moreover, we compare various combinations of these models to exploit their advantages. We also study the effect of diverse versions of pre-trained word embeddings.

The organization of this paper is as follows. After discussing prior work (section 2), we present the dataset used and describe the deep learning architectures studied in this work (section 3). Then, we evaluate the models and discuss their results (section 4). Finally, we provide some conclusions extracted from the experimentation (section 5).

2. Related work

Although sentiment analysis has been extensively applied to many application domains, the pharmaceutical domain has received much less attention. Early work in sentiment analysis of drug reviews mainly used rules[6] and sentiment lexicons (such as SentiWordNet[7]) [8, 9] to detect the overall polarity (positive or negative) of a given drug review.

A Bag of Word (BoW) approach was proposed by Bobicev et. al. [10] to represent twitter messages disclosing personal health information. The authors explored different machine learning algorithms such as Naive Bayes, Decision trees, KNN and SVM. In [11], several algorithms (such as Naive Bayes, SVM or Logistic Regression) were investigated to estimate the polarity of patients' posts in online health forums. The algorithms were trained using common sentiment analysis features such as the number of subjective words, the number

of adjectives, adverbs and pronouns, and the number of positive, negative and neutral words, taken from the Subjectivity Lexicon[12].

Mishra and colleagues[13] proposed a system based on Support Vector Machine (SVM) for detecting polarity of drug reviews. The system also performed aspect-based sentiment analysis on the drug reviews to predict ratings for some conditions such as satisfaction, effectiveness and ease of use of the drug. Drug reviews were tokenized. Then, SentiWordNet was used to assign the sentiment scores for each token.

Grasser et al.[14] created a dataset with drug reviews collected from the Drugs.com website, which provides information on drugs to both patients and health professionals. Each drug review includes a score from 0 to 9, which reflects the patient's degree of satisfaction with the drug. The reviews were grouped into three classes according their ratings: positive (rating>=7), negative (rating<=4) and neutral (rating in [3,6]). The authors used a logistic regression to classify the drug reviews, achieving an accuracy of 0.9224.

A word embedding model is a mapping between words and vectors capable to capture the similarity between words. Word embeddings generated by neural networks were first introduced by Bengio et. al. [15]. Since then, word embeddings have been widely and successfully used in various NLP tasks. Carrillo et. al. [16] were amongst the first to use word embeddings in sentiment analysis of patients' posts. The authors explored different machine learning algorithms such as SVM, Naive Bayes and Random Forest, which were trained using lexical, syntactic, semantic, sentiment analysis features and word embeddings to represent texts.

More recently, deep learning models have shown remarkable results for sentiment analysis in health domain. Yadav et. al. [17] compare the performance of a CNN with traditional machine learning algorithms, including SVM, Random Forest and a Multi-layer perceptron (MLP) to perform aspect-based sentiment analysis for aspects such as 'medication' and 'medical condition'. The CNN model achieved significant improvement compared to traditional algorithms.

Min [18] proposed a hybrid architecture combining the advantages of CNN

and Bi-LSTM to perform the binary classification (positive and negative) of drug reviews. The dataset consists of drug reviews that were gathered from the forum Askapatient[1]. Each review includes a rate (from 1 to 5) denoting the grade of user's satisfaction for a given drug. To alleviate the difficulty of the task, ratings were grouped in three classes (positive for 4 and 5 ratings, neutral for 3, and negative for 1 and 2 ratings).

Transfer learning is a new paradigm in machine learning, whose main idea is reusing knowledge learned for one task to solve other similar ones. In fact, the use of transformer models has begun to be widely used for NLP tasks such as text classification, question answering, and named entity recognition (NER) [19, 20]. In the last years, several approaches have been developed to obtain pre-training contextual representations, such as Semi-supervised Sequence Learning [21], ELMo[22], ULMFit[2]. Bidirectional Encoder Representations from Transformers (BERT) [23] is one of the most widely used transformers, whose main difference to the previous models is that BERT performs deep bidirectional (both left-to-right and right-to-left direction) representation from unlabeled text (wikipedia). BERT has shown to obtain new state-of-the-art results on several NLP tasks [4, 23].

Since last year, several systems have been developed using BERT for the sentiment analysis task. These works have provided state of the art results [24, 25, 26, 27], but have rarely focused on the sentiment analysis of drug reviews. Biseda and Mo [28] examined the performance of BERT on several tasks, including the sentiment analysis of drug reviews. They used the dataset created by Grasser et al. [14], which consists of drug reviews from drugs.com. Each drug review is related to a specific drug and has a rating (from 0 to 9) reflecting overall patient satisfaction. Biseda and Mo grouped the drug reviews into three polarities: highly negative (rating ≤ 3), highly positive (rating ≥ 8) and neutral (rating in [4,7]). Their experiments showed an accuracy of 0.906 for the sentiment analysis task.

[1] https://www.askapatient.com/

In sum, prior works suggest that deep learning has been rarely used in sentiment analysis of drug reviews.

3. Methods

3.1. Dataset

We use the dataset proposed in [14], which is a collection of drug reviews taken from the Drugs.com website. Each drug review includes a score from 0 to 9, which reflects the patient's degree of satisfaction with the drug. For example, a patient with *atrial fibrillation* posted the following comment: "*Only on it for 8 days. After 5 days started having shortness of breath, muscle spasms in upper back, pounding heart rate, fatigue, stiffness in neck and face*". This comment refers to the drug *Flecainide* and describes a series of adverse effects it has produced within a few days. The score of the drug provided by the patient is negative with a value of 1.

The corpus contains a total of 215,063 drugs reviews. The corpus is split into training and test datasets in the ratio 75:25, maintaining in both the proportion of the classes by stratified random sampling. Additionally, 15% of the total training dataset is used as development dataset, which allows us to learn the best hyperparameters of the different models.

Moreover, the creators of this corpus also grouped the drug reviews according three levels of polarity based on the review's rating: negative (class 0, rating<=4), neutral (class 1; rating in [3,6]) and positive (class 2; rating>=7).

In general, sentiment analysis is seen as a text classification task [29, 30, 4]. For this reason, we address the task using the three polarities proposed in [14], but also as a more challenging classification task using 10 classes. As mentioned before, each drug review is classified by an integer number (rating) from 0 to 9, which refers to the overall patient satisfaction. Thus, we consider these ratings as the classes for our problem.

For 10-class dataset, Figure 1 shows the distribution of the different drug reviews according to their class (degree of satisfaction). There is a strong un-

balanced distribution of the classes, predominating those with more polarized scores (such as 1, 8, 9 and 10).

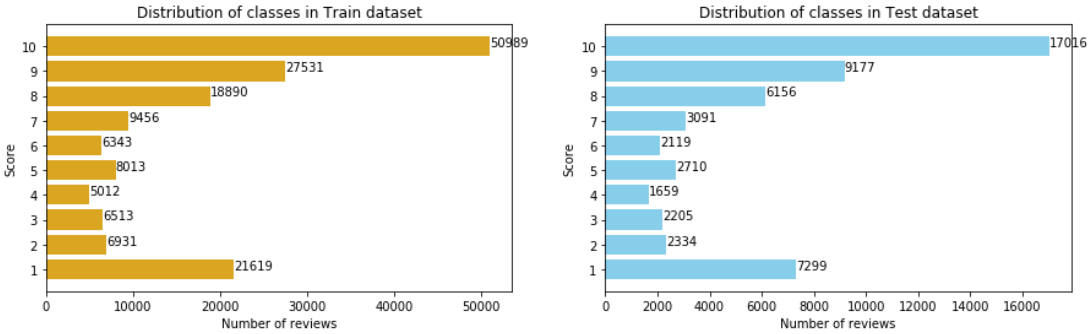


Figure 1: Class distribution in training and test datasets (10 classes)

The 3-class dataset also has an unbalanced distribution of the classes (see Figure 2). This figure shows that the distribution of the three classes in test and training sets is similar. The reviews with positive polarity (class 2) represent approximately 66%, while the reviews with positive polarity are around 25% and only 9% for the reviews with neutral polarity (class 1).

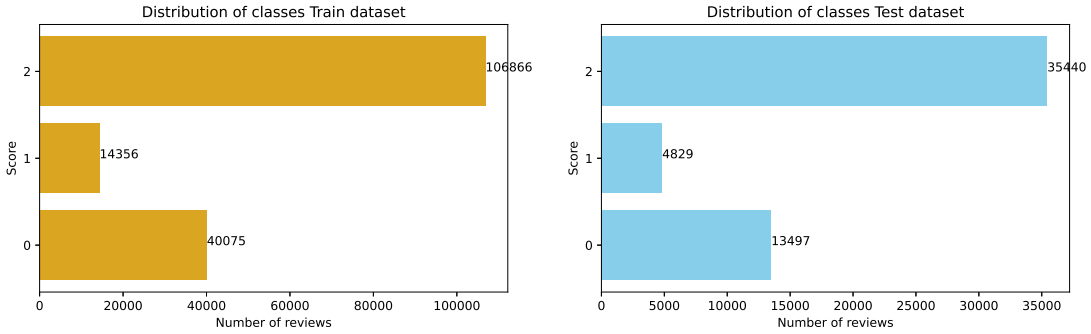


Figure 2: Class distribution in training and test datasets (3 classes)

3.2. Word embedding models

In this section, we describe the different pre-trained word embedding models that we will use to initialize our networks. These models, which were trained with Word2vec [31], are described below:

- The word embeddings model described in [32], which from now on will be called "WE model A", was trained on a collection of texts from PubMed, PMC, and Wikipedia in English (2013 version). This model contains a vocabulary of 5,443,656 words and the dimension of the word embeddings is 200.

- The word embeddings model presented in [33], which from now on will be called "WE model B", was trained using more than a million sentences in English from tweets about drugs. It contains a vocabulary of 26,278 lemmatized words and the dimension of the embeddings is 150.

- The word embeddings model described in [34], which from now on will be called "WE model C", was trained on a collection of twitter and Wikipedia (2017 version) texts in English, Reuters' news articles, the UMBC web-based corpus, and The "One Billion Word Language Modeling Benchmark". This model contains a vocabulary of 2,156,970 words and the dimension of the embeddings is 300.

To represent the drug reviews using these pre-trained word embedding models, we need to preprocess the reviews. The texts were tokenized and the numerical expressions blinded. Then, each text is represented as a matrix of word embeddings. When we employ the "WE model B" to represent the tokens, we also use the NLTK[2] lemmatizer to obtain as much vocabulary coverage as possible. The pre-processed drug reviews resulted in a vocabulary of 45,278 tokens when using lemmatization, and 49,339 tokens when omitting the lemmatization step.

Due to the different lengths of the drug reviews, it was necessary to pad and truncate the texts to equal their lengths. Based on the cumulative distribution function over the length of reviews (see Figure 3), almost 100% of processed texts have a length of less than or equal to 250 tokens (2,006 is the maximum length). Thus, we set the length of processed texts to 250 tokens.

[2]http://www.nltk.org/

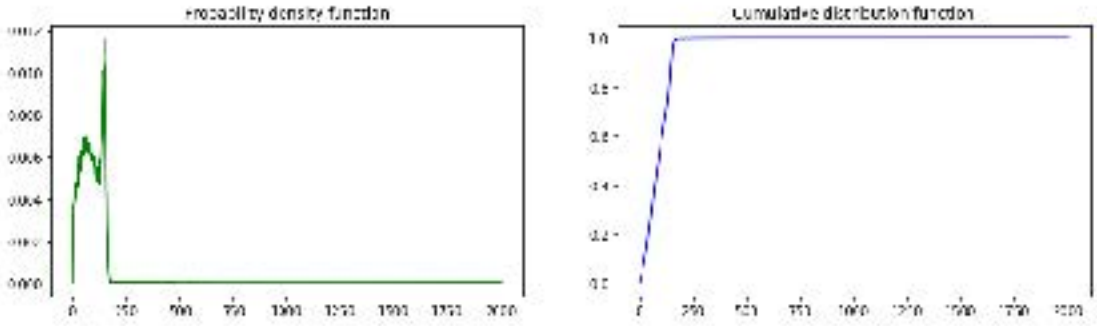


Figure 3: Probability density function and Cumulative distribution function over the length of reviews

3.3. Deep learning models

In this section, we describe several deep learning architectures for sentiment analysis of drug reviews. In particular, we propose different deep learning architectures: 1) a simple CNN, 2) a bidirectional LSTM, 3) a hybrid model comprising a CNN followed by a bidirectional LSTM, 4) a hybrid model including a CNN concatenated with a bidirectional LSTM, and 5) a hybrid model consisting of a bidirectional LSTM followed by a CNN. Moreover, we propose a sixth model based on BERT with a LSTM as classifier.

3.3.1. Simple models

The first of our deep learning approaches consists of a simple CNN architecture. This type of network can extract representative patterns that describe the text in the form of n-grams [35]. The CNN architecture has a convolution layer, where different filters operate sliding along the matrix of word embeddings of each drug review, producing as output a mapping of features of the reviews.

This architecture is composed of 64 filters with a window size of 2, 3 and 5-word vectors. We used a linear rectification unit (ReLU) [36] as activation function to avoid gradient vanishing problems [37]. Once the feature mapping is obtained, we employed a pooling based on the maximum of the values of each convolution. This method is used to prevent the padding operation from negatively affecting the representation of the texts [38].

Our second approach is based on a bidirectional LSTM, which processes the input text storing the semantics of the previous and future tokens. This type of recurrent network (RNN) is able of capturing contextual information and long-term dependencies[39]. LSTM layers are composed of recurrently connected memory blocks where each of the memory cells contains three multiplicative gates. These gates are able to use, store, and forget information for long periods, solving the vanishing gradient problem. Our bidirectional LSTM has a hidden state dimension of 250 for the forward and backward layers and hyperbolic tangent (tanh) as activation function.

3.3.2. Hybrid models combining CNN and LSTM

The following approaches consist of a combination of those described above. These combinations aim to capture both the local features of the texts and their global and temporal semantics [40, 18, 41]. The combination of these features may provide more discriminative information to perform the inference of the different classes.

For our third approach, we used the same CNN architecture described above to extract a sequence of representations of higher-level texts. These representations are conformed by the concatenation of the pooling layers and are used to feed a bidirectional LSTM to capture contextual information from the local features. For the convolutional layer, we used 64 filters with a window size of 2, 3 and 5-word vectors with ReLU activation and a max-pooling layer. For the bidirectional LSTM layer, we used a hidden state dimension of 200 for the forward and backward layers and hyperbolic tangent (tanh) as activation function.

For our fourth approach, we used a parallel combination of the CNN and bidirectional LSTM architectures described above to provide both contextual and local features from the text. With this parallel combination, we are able to extract and concatenate both types of features directly from the word embedding representation of the reviews. As can be seen in Figure 4, our architecture is composed of 64 filters with a windows size of 2, 3, and 5-word embeddings and a max-pooling layer for obtaining the local features. To obtain the contextual

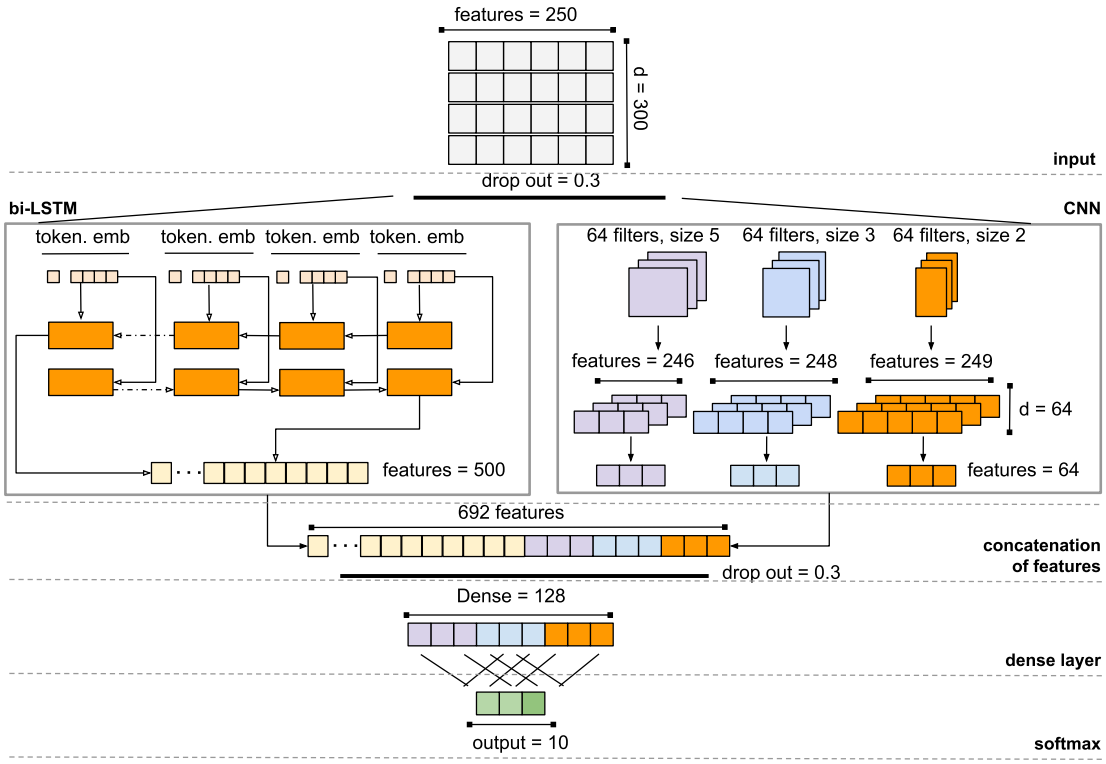


Figure 4: Overview architecture of the concatenation of the CNN and LSTM models

features our architecture is also composed of a bidirectional LSTM layer with a hidden state dimension of 250 for the forward and backward layers. We used a linear rectification unit (ReLU) and hyperbolic tangent (tanh) as activation function respectively.

For our last approach, we used an architecture with the aim of extracting local features from the abstract representation provided by a bidirectional LSTM layer. This abstract representation contains contextual information from the reviews. As can be seen in Figure 5, the word embedding of each review is used as input to a bidirectional-LSTM layer with a hidden state dimension of 250 for the forward and backward layers. This layer does not return only the sequence obtained in the last time step, but it returns the complete sequence of features with the information of each time step. Once the complete sequence

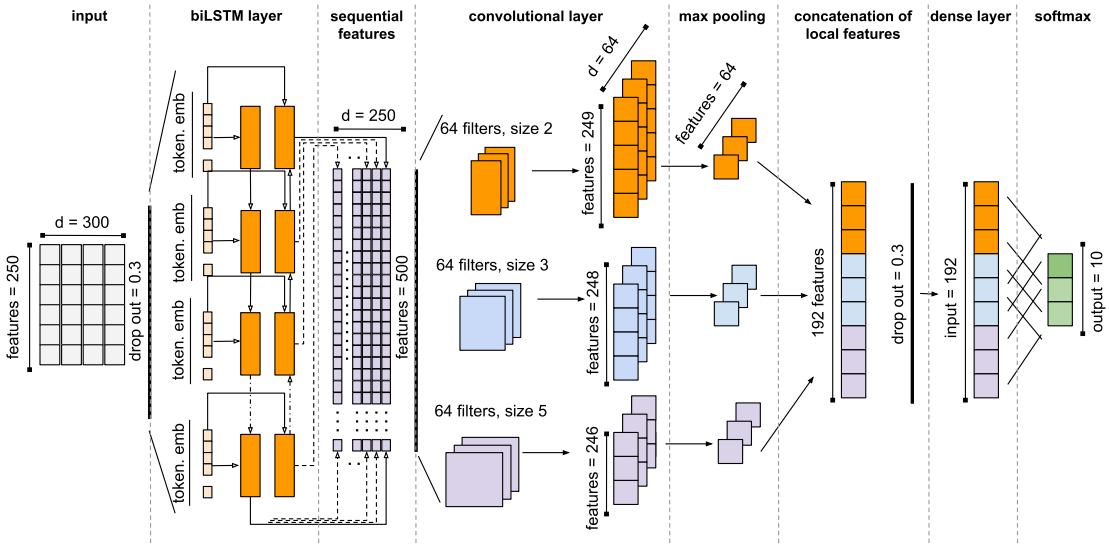


Figure 5: Overview architecture of our LSTM + CNN system

of hidden states is obtained, we used a convolutional layer to extract local information between the different time steps. To obtain the local features, 64 filters are employed with a window size of 2, 3 and 5 steps. A max-pooling layer is used to provide the most relevant features. These features, obtained by each type of convolutional filter, are finally concatenated. For the LSTM+CNN architecture, we used a linear rectification unit (ReLU) and a hyperbolic tangent (tanh) as activation functions for the convolutional and bidirectional-LSTM layers respectively.

Finally, the last layer of all the architectures described above is composed of fully connected perceptrons. In this layer, the ReLU activation function is used in order to improve class prediction. At this point, the different vectors representing each of the texts are connected to a Softmax layer, which predicts the score for each review.

For the training of the different models, we used the ADAM optimizer [42], with a learning rate of 0.001, a batch size of 200 reviews during 200 epoch and categorical cross-entropy as loss function.

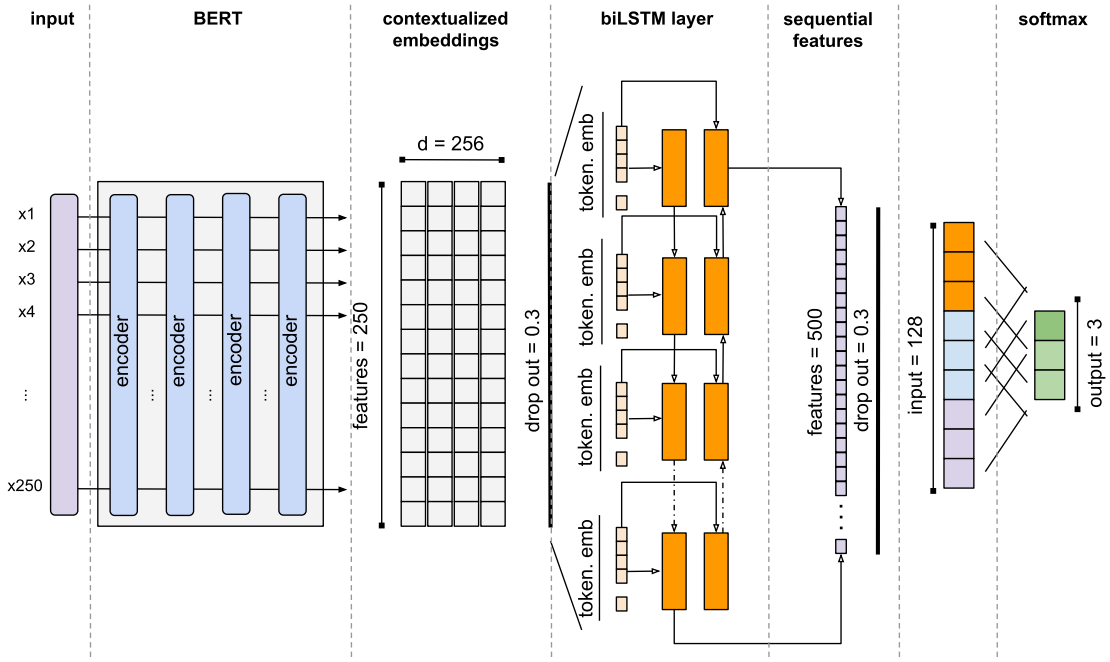


Figure 6: Overview architecture of BERT + LSTM model

3.3.3. BERT and LSTM model

We also explore the usage of BERT as a language model to represent the drug reviews. The main advantage of BERT compared to Word2Vec models is its ability to generate contextualized word embeddings. Thus, in a word2vec model, the word "*sentence*" will have the same representation for different sentences (for example, "*The judge gave a sentence*" and "*A sentence is a linguistic structure*"). However, if we use BERT, the representation of the word "*sentence*" will be different for those sentences. This is due to BERT generates contextualized word embeddings taking into account the context to provide the most accurate embedding. On the other hand, fine-tuning BERT for a specific task is inefficient from a parameter point of view. Due to the computational cost, we use a pre-trained BERT model with four encoder layers provided by Devlin et al. [23], which is known as BERT-small model.

Moreover, as a transfer mechanism, we use adapter modules proposed by

Houlsby et al. [43] to considerably reduce the number of trainable parameters. The use of adapter modules consists of injecting new layers into the original BERT model. Adapter modules are a bottleneck architecture that projects the input features to a smaller dimension m (we set m to 4 for our experimentation), applies non-linearity and projects the features to the original dimension. The adapter module also incorporates an internal skip-connection to avoid problems with near-zero initialization. The original BERT model weights remain unchanged during fine tuning for the specific task.

To represent the drug reviews based on BERT, the texts were tokenized using the vocabulary of BERT's pre-trained model and the tokenizer provided by bert-for-tf2[3]. The pre-processed texts resulted in a vocabulary of 30,522 tokens. As a last preprocessing step, based on the cumulative distribution function over the length of reviews (see Figure 3), we set the length of processed texts to 250 tokens.

Since the output of the BERT model consists of a representation of the texts using contextualized word embeddings, we can add top layers for the classification task. In particular, as can be seen in Figure 6, we use a bidirectional LSTM layer followed by a fully connected perceptron and softmax layers. We have choosen Bi-LSTM because it is simpler than the hybrid architectures and obtains similar results, as will be seen in Section 4.

For the training of the model, we used the ADAM optimizer [42], with a learning rate of 0.001, a batch size of 200 during 200 epoch and categorical cross-entropy as loss function.

Our source code is publicly available to enable the reproducibility of our experiments[4].

[3] https://pypi.org/project/bert-for-tf2/

[4] https://github.com/ccolonruiz/DrugSentimentAnalysis/

4. Results and Discussion

To evaluate our systems, we have used the standard metrics for text classification tasks: precision, recall and F1. These metrics can be extended to multi-classification problems using the micro-averaged and macro-averaged versions. In the macro-average, we show the mean of the values obtained for each class independently. In the micro-average, we add the contributions of each class and then compute the average metric. In general, micro-average is preferable for class imbalance problems.

4.1. Results obtained for 10 classes

As a baseline, we use a classical classifier Support Vector Machines (SVM) [44] with a tf-idf representation of the reviews, which has been proven very effective for text classification [45]. The parameter tuning was performed using grid search. This classifier obtains a macro-F1 of 41.68% and a micro-F1 of 51.97%. As expected, classes with a larger number of examples (0 and 9) tend to provide better results since the classifier has more examples to learn.

Table 2 shows the results of the CNN models. Models using random initialization or non-static word embeddings provide very similar performance with micro F1 around 66%-67% and macro F1 around 60%-62%. However, models using static word embeddings show low results, around 46%-48% of micro-F1 and 26%-30% of macro-F1. Comparing with the baseline system, while the static CNN models perform worse, the non-static CNN models provide significantly better results than the baseline system, with an improvement of almost 15 percentage points in micro-F1 and 20 points in macro-F1. Therefore, a CNN model trained with non-static word embeddings overcomes our SVM baseline.

With regard to the effect of the different word embedding models on the results, the non-static CNN model trained with the "WE model C" model (which was trained with a large collection of tweets and open-domain texts) achieves slightly better micro-F1 (67.4%) and macro-F1 (62.8%) than the other CNN model. On the other hand, random initialization also yields very similar results

to those provided by the non-static CNN models. Therefore, we can conclude that the word embedding models studied do not guarantee a significant improvement on the random initialization.

Linear SVM		
Label	**F1-score**	**Support**
0	0.6571	7299
1	0.3832	2334
2	0.3608	2205
3	0.3570	1659
4	0.3434	2710
5	0.3249	2119
6	0.3176	3091
7	0.3543	6156
8	0.3952	9177
9	0.6744	17016
Micro	0.5197	53766
Macro	0.4168	53766

Table 1: Results of Linear SVM (baseline)

LSTM far exceeds the baseline results (see Table 3). Unlike the CNN models where there is a strong difference (around 20 percentage points) between the performance of the static and non-static models, the non-static LSTM models show better results than the static LSTM ones, but the improvement is more modest (from 3 to 7 points). However, as happened in CNN models, there is no significant difference between the performance using random initialization and non-static word embeddings. In this case, the word embedding model that provides better results is the 'WE model A', which was trained with biomedical and general texts. This LSTM model achieves a micro-F1 of 69.1% and a macro-F1 of 63.9%. However, the differences between CNN and LSTM are not statistically significant.

	CNN models (F1-score)							
	Random		**WE model A**		**WE model B**		**WE model C**	
Label	**Non-Static**	**Static**	**Non-Static**	**Static**	**Non-Static**	**Static**	**Non-Static**	**Support**
0	0.7288	0.6474	0.74	0.6248	0.7355	0.6572	0.745	7299
1	0.6042	0.1655	0.5983	0.1485	0.5926	0.1598	0.6158	2334
2	0.6118	0.1684	0.5764	0.1801	0.5712	0.2287	0.608	2205
3	0.588	0.1224	0.6062	0.0996	0.5771	0.1448	0.6185	1659
4	0.5743	0.1886	0.5947	0.1886	0.5577	0.2331	0.5881	2710
5	0.5613	0.1247	0.5462	0.1112	0.5517	0.1452	0.5777	2119
6	0.5647	0.1431	0.562	0.1096	0.5518	0.1952	0.5626	3091
7	0.5869	0.2754	0.5843	0.2387	0.575	0.2424	0.597	6156
8	0.6194	0.3016	0.6098	0.3098	0.605	0.3508	0.6129	9177
9	0.7585	0.6807	0.7591	0.6701	0.7589	0.6908	0.7636	17016
micro avg	0.6681	0.475	0.6685	0.4618	0.6616	0.4871	**0.6745**	53766
macro avg	0.6198	0.2818	0.6177	0.2681	0.6077	0.3048	**0.6289**	53766

Table 2: Results of the CNN models using different initializations. The best results are shown in bold

Table 4 shows the results for the three hybrid architectures combining CNN and LSTM. We use random initialization for the three models since none of the word embedding models offer significantly better results than random initialization. Our results show that LSTM followed by a CNN obtains the highest micro-F1 and macro-F1, however, the differences with respect to the other hybrid architectures are not statistically significant.

As previously described, there is a strong unbalanced distribution of the classes, predominating those with more polarized rating. This results in a challenge when we try to classify those reviews with less representation. The analysis of the results reveals that the performance of each class is remarkably dependent of the number of instances of each in the training set, regardless of the model used. So the classes with higher number of training instances usually

	LSTM models (F1-score)							
	Random		**WE model A**		**WE model B**		**WE model C**	
Label	**Non-Static**	**Static**	**Non-Static**	**Static**	**Non-Static**	**Static**	**Non-Static**	**Support**
0	0.7575	0.7799	0.7732	0.7499	0.7665	0.7825	0.7687	7299
1	0.6077	0.5333	0.6249	0.4859	0.6161	0.532	0.6374	2334
2	0.6181	0.5048	0.6058	0.4511	0.6057	0.5239	0.5943	2205
3	0.5953	0.5019	0.5836	0.4066	0.605	0.4839	0.6142	1659
4	0.5848	0.5108	0.6017	0.4592	0.5829	0.5052	0.5764	2710
5	0.5743	0.463	0.5909	0.3863	0.5897	0.4171	0.57	2119
6	0.5786	0.4941	0.5802	0.4252	0.5734	0.4755	0.5749	3091
7	0.5939	0.5675	0.6074	0.4957	0.5997	0.5461	0.5968	6156
8	0.635	0.5988	0.6376	0.5489	0.6398	0.5881	0.6409	9177
9	0.7783	0.7904	0.7858	0.7688	0.7853	0.7905	0.782	17016
micro avg	0.6835	0.6583	**0.6918**	0.616	0.6886	0.6518	0.6871	53766
macro avg	0.6323	0.5744	**0.6391**	0.5178	0.6364	0.5645	0.6356	53766

Table 3: Results of the LSTM models using different initializations. The best results are shown in bold

show better results than those with less number of instances in the training set.

Approaches based on CNN architectures with pre-trained static embeddings provide the lowest results of all our experimentation, especially in the under-represented classes. CNN networks are good at extracting local and location-independent features, but they are not able to extract information from long-range semantic dependencies [39]. The reason that LSTM models with pre-trained static embeddings perform better in our experimentation might be due to long dependencies contained in the texts of the reviews (see Table 3). For example, this review: "*I was taking* Aviane *before and transferred my prescription to a different pharmacy. There they gave me* lutera *since they did not have* aviane*... Worst experience of my life. Constant headaches and acne were the main symptoms right before my period. Extremely painful periods as well.*

	Hybrid models (F1-score)			
Label	**CNN concat LSTM**	**CNN + LSTM**	**LSTM + CNN**	**Support**
0	0.7625	0.761	0.7646	7299
1	0.6187	0.648	0.6268	2334
2	0.6161	0.4784	0.6068	2205
3	0.5916	0.6444	0.613	1659
4	0.5926	0.5591	0.587	2710
5	0.5653	0.5968	0.5966	2119
6	0.5745	0.6112	0.5739	3091
7	0.6005	0.6071	0.6092	6156
8	0.6266	0.6381	0.6443	9177
9	0.7756	0.7707	0.7865	17016
micro avg	0.6825	0.68	**0.6928**	53766
macro avg	0.6324	0.6315	**0.6409**	53766

Table 4: Results of the hybrid architectures (using random initialization). The best results are shown in bold

If you are experiencing these symptoms while on lutera*, I would highly recommend trying* aviane*. I do not have any headaches or acne and my periods are pain free.*". It refers to the side effects produced by the drug *Lutera*, but it also recommended using *Aviane* as a substitute for *Lutera*. The rating predicted by the CNN network is highly positive (maximum value), while the LSTM network predicts a negative rating of 2 (correct rating). However, when we remove the part of the text that indicates the good performance of *Aviane*, the CNN network scores the review with a negative value of 1 (minimum value). In this case, the LSTM network is able to maintain information from the context of the full review, where the assessment is negative towards the drug *Lutera*.

On the other hand, allowing the adjustment of the word embeddings during the training time, provides a significant improvement, especially in the case of CNN classifiers (see Table 2). This improvement might be due, among other

reasons, to the vocabulary coverage provided by the different pre-trained embedding models. For example, the "WE model A" vocabulary contains 69.92% of the dataset vocabulary, the "WE model B" model contains only 30.54% and the "WE model C" model covers 78.23%. Terms not represented by these embedding models do not provide semantic information to the classifiers. The fine-tuning of the embeddings allows adjusting their values to the context of the problem by modifying the previously learned associations (under the risk of overfitting). In this way, those terms that did not provide information are differentiated from each other, fitting to the problem.

Finally, hybrid approaches that combine different CNN and LSTM network architectures (allowing the adjustment of the word embeddings) are used to obtain both local and sequential features. The resulting classifiers provide slightly higher results than those provided by the non-static CNN models, but similar to the provided by the non-static LSTM ones (the difference in micro-F1 and macro-F1 averages ranges from 1% as shown in Tables 3 and 4). As we have discussed above, this might be due to providing the new models with the ability to maintain long dependency information contained in the texts.

Despite this, our approaches are not always able to accurately classify those reviews where the text contains sarcasm, irony or humor. For example, in the following review we can observe all these elements: "*My Dr thought this would be a good idea for me because of my knee condition I had had for years. I figured I'd give it a go and big surprise it made it worse a lot worse - it was like he stirred up a hornets nest after those shots. Now I'm having more trouble with it than ever before with walking pain, sleeping pain, knee resting pain, it's all worse I'm 40 and walk like I'm 80. Physical therapy was a joke considering my condition can't be fixed without replacing the knee. All physical therapy did was get me to shift my weight to one side and now my other knee is giving me problems. I guess the Dr won't be happy until I'm in a wheelchair*". The classifiers LSTM and CNN concatenated with LSTM correctly score the example (rating of 0), but the rest of the classifiers score with very different values (CNN: 7, CNN+LSTM: 6, LSTM+CNN: 5). In this case, all our classifiers correctly score the example

by removing expressions such as: "*it was like he stirred up a hornets nest after those shots*" and "*I guess the Dr won't be happy until I'm in a wheelchair.*"

Overall, the top performance is a micro-F1 of 69.28% and a macro-F1 of 64.09% provided by the LSTM+CNN hybrid classifier (see Table 4). This classifier aims to extract local features with the convolutional layer from the contextual information provided by the bidirectional LSTM layer. This might be since the sequential features obtained directly from the representation of embeddings (by the bi-directional LSTM layer) are the most discriminating ones in our experimentation. Providing local information on these features might improve the performance compared to classifiers based only on LSTM networks.

However, we should have to take into account the noise in the data sets based on subjectivity in the grade of satisfaction. Our task is extremely difficult especially considering the number of possible choices (10) and the subjectivity differences especially those between classes very close to one another. This results in models that can accurately distinguish between positive and negative polarities, but having difficulty in providing the accurate rating. Thus, when we classify a review (whose rating is 10), we should distinguish different types of errors, because a predicted rating of 9 or 8 should be considered more accurate than 5 or a lower value. For example, the following review is only correctly labeled by the CNN+LSTM classifier (with a value of 7): "*I started taking gabapentin experimentally to treat chronic depression and am now prescribed 1800mg a day. At doses of 2700mg I experienced a significant improvement in impulse control and depression in general. Went from being a shut in, to going for walks and enjoying my time in the company of others. At the lower dose of 1800mg a day, I don't experience much improvement, save for a decrease in depressive symptoms. I'm looking forward to a larger dose if my doctor okays it.*" The remaining classifiers, however, label it with very similar degrees of satisfaction, for example, 6 or 8.

Another factor to take into consideration is the time needed for the training of each of the models (see Table 5). Our experimentation was done with an Nvidia Titan XP graphics card, an Intel Core i7-6700K and 32GB of RAM at

	Training time (minutes)						
	WE model A		WE model B		WE model C		Random
Model	**Non-Static**	**Static**	**Non-Static**	**Static**	**Non-Static**	**Static**	**Non-Static**
CNN	43'	30'	36'	23'	60'	36'	43'
LSTM	1383'	800'	870'	796'	906'	833'	886'
CNN+LSTM	-	-	-	-	-	-	710'
LSTM+CNN	-	-	-	-	-	-	950'
CNNconcatLSTM	-	-	-	-	-	-	916'

Table 5: Training time in minutes for 200 epochs using the Keras 2.0 library (10-class dataset)

1600 MHz. The training time using the Keras 2.0 library for LSTM models (200 epochs) far exceeds the time needed to train CNN models (the training time for LSTM models is approximately 20 times longer).

4.2. Results obtained for 3 classes (positive, negative and neutral)

In this subsection, we focus on the experiments performed using the dataset with three polarities: positive, neutral and negative. As seen Tables 6, 7, 8 and 9, all of our models show much better performance compared to the models evaluated on the dataset with ten classes. Thus, the highest micro-F1 (0.6928) and macro-F1 (0.6409) (see Table 4), which were provided by the CNN+LSTM model, are almost 20% lower than those provided by the same model evaluated on the dataset with three polarities. This may be due to the number of instances for each class is increased when we work only with three classes.

Table 6 shows the results of the baseline system (Linear SVM) on the dataset with three polarities with a tf-idf representation of the reviews. Table 7 shows the results of the simple models (CNN and LSTM) on the data set with three polarities. CNN and LSTM provide very similar performance, as happened on the dataset with ten classes (see Tables 2 and 3). The simple models CNN and LSTM provide significantly better results than the baseline system, with

an improvement of almost 7 percentage points in micro-F1 and 14 points in macro-F1

Linear SVM		
Label	**F1-score**	**Support**
0	0.7430	13497
1	0.3579	4829
2	0.8892	35440
micro avg	0.8175	53766
macro avg	0.6634	53766

Table 6: Results of Linear SVM with tf-idf representation (baseline)

CNN and LSTM models (F1-score)			
Label	**CNN**	**LSTM**	**Support**
0	0.8364	0.8482	13497
1	0.6457	0.6361	4829
2	0.9302	0.9342	35440
micro avg	0.8844	**0.8882**	53766
macro avg	0.8041	**0.8062**	53766

Table 7: Results of the CNN/LSTM architectures (using random initialization). The best results are shown in bold

The hybrid models combining CNN and LSTM (see Table 8) show slightly better performance than the simple models, with an increase of 0.62% on the micro-F1 score and 0.84% on the macro-F1 score.

Recently, BERT models have shown huge improvements in a wide range of NLP tasks. However, as seen in Table 9, BERT embeddings with LSTM provide slightly better results compared to the previous models. The model achieves

	Hybrid models (F1-score)			
Label	**CNN concat LSTM**	**CNN + LSTM**	**LSTM + CNN**	**Support**
0	0.8498	0.8560	0.8545	13497
1	0.6398	0.6507	0.6393	4829
2	0.9359	0.9371	0.9372	35440
micro avg	0.8903	**0.8944**	0.8920	53766
macro avg	0.8085	**0.8146**	0.8103	53766

Table 8: Results of the hybrid architectures (using random initialization). The best results are shown in bold

	BERT + LSTM model	
Label	**F1-score**	**Support**
0	0.8720	13497
1	0.6514	4829
2	0.9477	35440
Micro	**0.9046**	53766
Macro	**0.8237**	53766

Table 9: Results of BERT + LSTM architecture. The best results are shown in bold

1.64 improvement on micro-F1 and 1.75 macro-F1 score over the simple LSTM model with random initialization. Compared to the hybrid models combining CNN and LSTM, the improvement is more modest, with only 1.02% on micro-F1 score and 0.91% on macro-F1 score.

Moreover, our system based on BERT has a similar accuracy (0.9046) as the previous work [28], which also exploited BERT. However, our system is not able to overcome the logistic regression model proposed in [14], whose accuracy was 0.9224.

Clearly, all of our models show a very similar behavior for the three classes

(see Figure 7). Thus, the drug reviews with positive polarity (class 2) are classified with high performance by all models. The top F1-score (0.9477) is achieved by the BERT embedding with LSTM. This model also achieves the highest F1-score (0.8720) for the reviews with negative polarity (class 0). Regarding the reviews with neutral polarity (class 1), all models show lower results than in the other classes, with the highest F1-score around 0.65. This may be due to this class only represents 10% of all instances.

In the confusion matrix (see Figure 7), we can see there are a large number of cases, for all models, where neutral class examples are classified as positive or negative. Moreover, also a large number of positive or negative class examples are classified as neutral class ones. This is conditioned by the small representation of neutral class examples. On the other hand, we can see that the BERT+Bi-LSTM model classifies fewer examples in opposite classes to the correct ones than the rest of the models, being the CNN model the one that makes more mistakes of this type. This may be due to the fact that the CNN model is not capable of storing information about long dependencies and does not have word-level contextualized information.

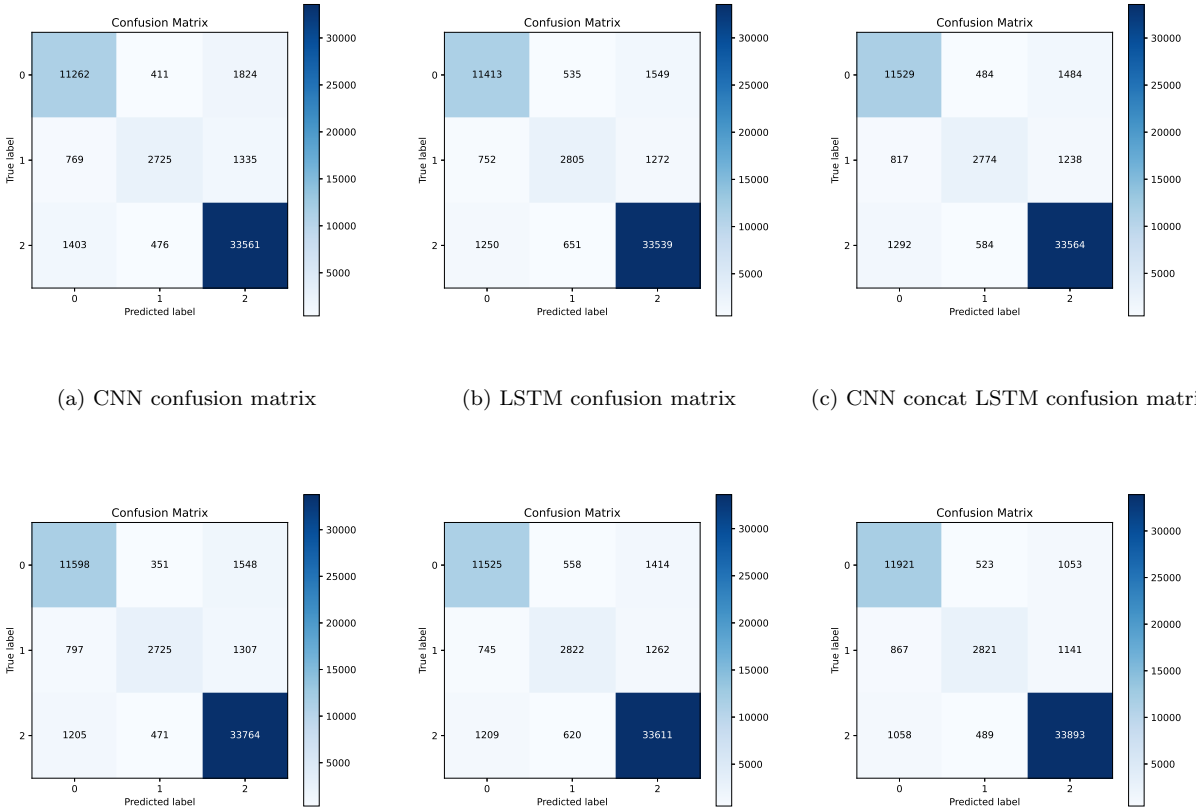
Training time (minutes)

Model	**Time**
CNN	217'
LSTM	353'
CNN+LSTM	333'
LSTM+CNN	953'
CNNconcatLSTM	953'
BERT+LSTM	1,343'

Table 10: Training time in minutes for 200 epochs using the Tensorflow 2.3.0-tf library (3-class dataset)

Regarding the training times for the models classifying three classes, we can see that the simple CNN is the model requiring less training time, while BERT

has the highest training time.



(a) CNN confusion matrix

(b) LSTM confusion matrix

(c) CNN concat LSTM confusion matrix

(d) CNN + LSTM confusion matrix

(e) LSTM + CNN confusion matrix

(f) BERT + LSTM confusion matrix

Figure 7: Model confusion matrices (3 classes)

5. Conclusion

This is the first work, to our best knowledge, to compare deep learning architectures for the task of sentiment analysis of drug reviews. We also study the effect of different word embedding models on the performance of the different models, however, none of them seems to offer significant better results than the

rest of word embedding models or random initialization.

Sentiment analysis is usually considered as a text classification task. In our case, we address the task using the three polarities (positive, negative and neutral) proposed in [14], but also as a more challenging classification task using 10 classes, which are the ratings defined by the users to show their overall satisfaction with the drug of the review. As is to be expected, the results for the 3-class dataset are much higher than those obtained with the 10-class dataset. In the 3-class dataset, there is not only less classes to classify, but also each class has more examples to learn.

Our results on the 10-class dataset show that the hybrid model composed of a bidirectional LSTM followed by a CNN provides the best results. CNN models initialized with static word embedding show very low performance for classes with fewer instances in the training set. On the other hand, the CNN model requires less training time than the bidirectional LSTM and hybrid models.

In the last two years, the irruption of BERT has revolutionized the NLP field, achieving state-of-art results for many NLP tasks. Thus, we also use BERT to represent our drug reviews and apply a Bi-LSTM to classify them. Focusing on the results for the 3-class dataset, we can observe that BERT followed by a Bi-LSTM provides slightly better results than the other models. However, using BERT considerably increases the computational cost. CNN provides acceptable results while requiring less training time.

Due to the increasing use of language representation models for classification tasks, as future work, we plan to apply them in addition to variational autoencoders (VAE) and adversarial networks for semi-supervised approaches. In this way, we plan to explore new methods in order to reduce the dependence on annotated corpora. In addition, we will explore the use of semantic features along with contextual ones in order to improve the fine-tuning of our approaches.

Acknowledgments

This work was supported by the Research Program of the Ministry of Economy and Competitiveness - Government of Spain, (DeepEMR project TIN2017-87548-C2-1-R) and the Interdisciplinary Projects Program for Young Researchers at Universidad Carlos III of Madrid founded by the Community of Madrid (NLP4Rare-CM-UC3M). .

References

[1] R. Johnson, T. Zhang, Supervised and semi-supervised text categorization using lstm for region embeddings, in: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML'16, 2016, pp. 526–534.

[2] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2018, pp. 328–339.

[3] B. N. Patro, V. K. Kurmi, S. Kumar, V. Namboodiri, Learning semantic sentence embeddings using sequential pair-wise discriminator, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 2715–2729.

[4] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, in: Advances in neural information processing systems, 2019, pp. 5754–5764.

[5] S. M. Jiménez-Zafra, M. T. Martín-Valdivia, M. D. Molina-González, L. A. Ureña-López, How do we talk about doctors and drugs? sentiment analysis in forums expressing opinions for medical domain, Artificial intelligence in medicine 93 (2019) 50–57.

[6] J.-C. Na, W. Y. M. Kyaing, C. S. Khoo, S. Foo, Y.-K. Chang, Y.-L. Theng, Sentiment classification of drug reviews using a rule-based linguistic approach, in: International conference on asian digital libraries, Springer, 2012, pp. 189–198.

[7] A. Esuli, F. Sebastiani, Sentiwordnet: A publicly available lexical resource for opinion mining., in: LREC, Vol. 6, Citeseer, 2006, pp. 417–422.

[8] L. Goeuriot, J.-C. Na, W. Y. Min Kyaing, C. Khoo, Y.-K. Chang, Y.-L. Theng, J.-J. Kim, Sentiment lexicons for health-related opinion mining, in: Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium, ACM, 2012, pp. 219–226.

[9] M. T. Wiley, C. Jin, V. Hristidis, K. M. Esterling, Pharmaceutical drugs chatter on online social networks, Journal of biomedical informatics 49 (2014) 245–254.

[10] V. Bobicev, M. Sokolova, Y. Jafer, D. Schramm, Learning sentiments from tweets with personal health information, in: Canadian conference on artificial intelligence, Springer, 2012, pp. 37–48.

[11] T. Ali, D. Schramm, M. Sokolova, D. Inkpen, Can i hear you? sentiment analysis on medical forums, in: Proceedings of the sixth international joint conference on natural language processing, 2013, pp. 667–673.

[12] T. Wilson, J. Wiebe, P. Hoffmann, Recognizing contextual polarity in phrase-level sentiment analysis, in: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Vancouver, British Columbia, Canada, 2005, pp. 347–354.
URL https://www.aclweb.org/anthology/H05-1044

[13] A. Mishra, A. Malviya, S. Aggarwal, Towards automatic pharmacovigilance: Analysing patient reviews and sentiment on oncological drugs, in:

2015 IEEE International Conference on Data Mining Workshop (ICDMW), IEEE, 2015, pp. 1402–1409.

[14] F. Gräßer, S. Kallumadi, H. Malberg, S. Zaunseder, Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning, in: Proceedings of the 2018 International Conference on Digital Health, ACM, 2018, pp. 121–125.

[15] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, Journal of machine learning research 3 (Feb) (2003) 1137–1155.

[16] J. Carrillo-de Albornoz, J. R. Vidal, L. Plaza, Feature engineering for sentiment analysis in e-health forums, PloS one 13 (11) (2018) e0207996.

[17] S. Yadav, A. Ekbal, S. Saha, P. Bhattacharyya, Medical sentiment analysis using social media: towards building a patient assisted system, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), 2018.

[18] Z. Min, Drugs reviews sentiment analysis using weakly supervised model, in: 2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), IEEE, 2019, pp. 332–336.

[19] D. Sarkar, R. Bali, T. Ghosh, Hands-On Transfer Learning with Python: Implement advanced deep learning and neural network models using TensorFlow and Keras, Packt Publishing Ltd, 2018.

[20] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., Transformers: State-of-the-art natural language processing, arXiv preprint arXiv:1910.03771.

[21] A. M. Dai, Q. V. Le, Semi-supervised sequence learning, in: Advances in neural information processing systems, 2015, pp. 3079–3087.

[22] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of NAACL-HLT 2018, Association for Computational Linguistics, 2018, pp. 2227–2237.

[23] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of NAACL-HLT 2019, Association for Computational Linguistics, 2019, pp. 4171–4186.

[24] H. Xu, B. Liu, L. Shu, P. S. Yu, Bert post-training for review reading comprehension and aspect-based sentiment analysis, in: Proceedings of NAACL-HLT 2019, Association for Computational Linguistics, 2019, pp. 2324–2335.

[25] C. Sun, L. Huang, X. Qiu, Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence, in: Proceedings of NAACL-HLT 2019, Association for Computational Linguistics, 2019, pp. 380–385. doi:10.18653/v1/N19-1035.
URL https://www.aclweb.org/anthology/N19-1035

[26] X. Li, X. Fu, G. Xu, Y. Yang, J. Wang, L. Jin, Q. Liu, T. Xiang, Enhancing bert representation with context-aware embedding for aspect-based sentiment analysis, IEEE Access 8 (2020) 46868–46876.

[27] Y. Song, J. Wang, Z. Liang, Z. Liu, T. Jiang, Utilizing bert intermediate layers for aspect based sentiment analysis and natural language inference, arXiv preprint arXiv:2002.04815.

[28] B. Biseda, K. Mo, Enhancing pharmacovigilance with drug reviews and social media, arXiv preprint arXiv:2004.08731.

[29] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, in: Advances in neural information processing systems, 2015, pp. 649–657.

[30] C. Sun, X. Qiu, Y. Xu, X. Huang, How to fine-tune bert for text classification?, in: China National Conference on Chinese Computational Linguistics, Springer, 2019, pp. 194–206.

[31] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality (2013). arXiv:1310.4546.

[32] S. Pyysalo, F. Ginter, H. Moen, T. Salakoski, S. Ananiadou, Distributional semantics resources for biomedical text processing, Proceedings of Languages in Biology and Medicine.

[33] A. Nikfarjam, A. Sarker, K. O'Connor, R. Ginn, G. Gonzalez, Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features, Journal of the American Medical Informatics Association 22 (3) (2015) 671–681.

[34] Q. Li, S. Shah, X. Liu, A. Nourbakhsh, Data sets: Word embeddings learned from tweets and general data, in: Eleventh International AAAI Conference on Web and Social Media, 2017.

[35] J. Wang, Z. Wang, D. Zhang, J. Yan, Combining knowledge with deep convolutional neural networks for short text classification, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, 2017, pp. 2915–2921. doi:10.24963/ijcai.2017/406.
URL https://doi.org/10.24963/ijcai.2017/406

[36] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th international conference on machine learning (ICML-10), 2010, pp. 807–814.

[37] H. Ide, T. Kurita, Improvement of learning for cnn with relu activation by sparse regularization, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 2684–2691.

[38] V. Suárez-Paniagua, I. Segura-Bedmar, Evaluation of pooling operations in convolutional architectures for drug-drug interaction extraction, BMC bioinformatics 19 (8) (2018) 209.

[39] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification, in: Twenty-ninth AAAI conference on artificial intelligence, 2015.

[40] J. Bao, L. Zhang, B. Han, Collaborative attention network with word and n-gram sequences modeling for sentiment classification, in: International Conference on Artificial Neural Networks, Springer, 2019, pp. 79–92.

[41] I. Segura-Bedmar, P. Raez, Cohort selection for clinical trials using deep learning models, Journal of the American Medical Informatics Association 26 (11) (2019) 1181–1188.

[42] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[43] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, S. Gelly, Parameter-efficient transfer learning for NLP, in: Proceedings of the 36th International Conference on Machine Learning, Vol. 97, 2019, pp. 2790–2799.

[44] V. Vapnik, V. Vapnik, Statistical learning theory (1998).

[45] T. Joachims, Text categorization with support vector machines: Learning with many relevant features, in: European conference on machine learning, Springer, 1998, pp. 137–142.