



# Article Towards Sentiment Analysis for Romanian Twitter Content

Dan Claudiu Neagu <sup>1,2,\*</sup>, Andrei Bogdan Rus <sup>1,3,\*</sup>, Mihai Grec <sup>1</sup>, Mihai Augustin Boroianu <sup>1</sup>, Nicolae Bogdan <sup>1</sup> and Attila Gal <sup>1</sup>

- <sup>1</sup> Cicada Technologies, Bd. Nicolae Titulescu 18 apt. 82, 400420 Cluj-Napoca, Cluj, Romania
- <sup>2</sup> Business Informatics Research Center, Babes-Bolyai University, Str. Theodor Mihali 58-60, 400591 Cluj-Napoca, Cluj, Romania
- <sup>3</sup> Faculty of Electronics Telecommunications and Information Technology, Technical University of Cluj-Napoca, Str. Memorandumului 28, 400114 Cluj-Napoca, Cluj, Romania
- \* Correspondence: dann@cicadatech.eu (D.C.N.); bogdanr@cicadatech.eu (A.B.R.)

Abstract: With the increased popularity of social media platforms such as Twitter or Facebook, sentiment analysis (SA) over the microblogging content becomes of crucial importance. The literature reports good results for well-resourced languages such as English, Spanish or German, but open research space still exists for underrepresented languages such as Romanian, where there is a lack of public training datasets or pretrained word embeddings. The majority of research on Romanian SA tackles the issue in a binary classification manner (positive vs. negative), using a single public dataset which consists of product reviews. In this paper, we respond to the need for a media surveillance project to possess a custom multinomial SA classifier for usage in a restrictive and specific production setup. We describe in detail how such a classifier was built, with the help of an English dataset (containing around 15,000 tweets) translated to Romanian with a public translation service. We test the most popular classification methods that could be applied to SA, including standard machine learning, deep learning and BERT. As we could not find any results for multinomial sentiment classification (positive, negative and neutral) in Romanian, we set two benchmark accuracies of  $\approx$ 78% using standard machine learning and  $\approx$ 81% using BERT. Furthermore, we demonstrate that the automatic translation service does not downgrade the learning performance by comparing the accuracies achieved by the models trained on the original dataset with the models trained on the translated data.

**Keywords:** natural language processing; sentiment analysis; underrepresented language; machine learning; deep learning; Twitter

# 1. Introduction

Nowadays, social media platforms such as Twitter, Facebook or Instagram have become a common place for people to share their thoughts and for media outlets to spread their information to the world. Supervising social media becomes effective as a way of surveillance, as the number of their users has rocketed around the world: by the end of 2021, the number of social media users was over 4.2 billion, and this number is projected to increase to 5.4 billion by 2025 [1]. A mean toward this is analyzing people's opinion, sentiments, and attitudes from text sources, also known as sentiment analysis (SA) [2]. Falling under the broad umbrella of text classification, SA systematically identifies, extracts, quantifies and studies affective states and subjective information, producing useful knowledge from text, to be used in subsequent descriptive or causal analysis [3].

Due to the high popularity of social media, platforms such as Twitter have become a good source for investigating public opinions. By mining such texts, private entities can employ methods such as brand monitoring in order to capture important brand events in real time, and reflect a brand's financial value to the firm [4]. Public entities, such as governments, can use sentiment analysis or emotion detection in order to observe



Citation: Neagu, D.C.; Rus, A.B.; Grec, M.; Boroianu, M.A.; Bogdan, N.; Gal, A. Towards Sentiment Analysis for Romanian Twitter Content. *Algorithms* **2022**, *15*, 357. https:// doi.org/10.3390/a15100357

Academic Editors: Melania Susi and Alwin Poulose

Received: 30 August 2022 Accepted: 24 September 2022 Published: 28 September 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). how the population is reacting to various public issues such as politics or healthcare. For example, Praveen et al. [5] analyzed the attitude of Indian citizens towards COVID-19 vaccines by collecting public social media posts written in English. They concluded that, although positive sentiments were more prevalent than negative ones, the Indian government needs to focus especially on addressing the fear of vaccines, which is a major factor contributing to the negative attitude towards vaccination. Even more knowledge can be extracted by modeling discussions on social networks. Bonifazi et al. [6] proposed a general multilayer network approach and proved its validity by applying it on a Twitter dataset containing texts concerning opinions on COVID-19 vaccines (pro-vaccination, neutral and anti-vaccination). They discovered that anti-vaxxers tend to have ego networks denser and more cohesive than those of pro-vaxxers, which leads to a larger number of interactions among anti-vaxxers.

SA represents an established category of natural language processing (NLP), with a lot of research efforts being focused on discovering the machine learning (ML) methods that produce the best models given a particular problem under study. Although textbooks such as [2] or reviews such as [3,7,8] present in details the recommended steps to be adopted specifically for SA or with respect to a given technology applied to NLP in general and SA in particular, a lot of research space is still open in the area of SA, if certain problem-specific conditions occur, such as those induced within microblogging platforms, or handling user input from mobile devices, etc.

Microblogging platforms such as Twitter, Instagram, Facebook, etc., inspire provocative questions as they feature linguistic challenges usually not seen in literary texts. Eisenstein [9] describes them as being *bad language* and it includes emoticons, phrasal abbreviations such as *lol, smh* or *ikr*, expressive lengthening of words, e.g., *cooolll*, shortened words, or simply words written in a non-standard format, including those with typos, selected from some irregular vocabulary or following an informal grammar. Various reasons not in the scope of our research cause the presence of bad language in microblogging and it dramatically influences the performance of a standard NLP model applied here [10]. Another issue regarding social media networks is related to privacy threats. The goal of any social network is to enable users to safely share information, but in most cases, the users are not familiar with privacy preservation issues. The work of Cerruto et al. [11] demonstrates that it is possible to obtain or reconstruct personal user data by single-social or cross-social network analysis. In our work, we aim to infer global sentiment polarity for social media texts and we want to clarify that our analysis is not aimed at violating or exploiting user privacy.

Specifically targeting Twitter, performing SA on non-English tweets is seen as challenging, mostly because of the difficulty to gather enough labeled data in the target language [12]. Annotated datasets could be found for popular languages in the world. For example, for English, we can mention BERTweet [10], a large-scale language model trained over a corpus of 850M Tweets, which could be used together with fairseq [13] or transformers [14] for text categorization tasks, including SA. In France, DEFT challenges run between 2014 to 2018 focused on opinion mining and SA from Twitter posts [15], and they gathered a labeled dataset at the disposal of the participating teams. In Spain, TASS workshop (http://tass.sepln.org/, accessed on 22 August 2022) held starting with 2012 at SEPLN congress supplies a dataset with annotated tweets [16], including Spanish crosslingual variations.

However, little could be found for less popular languages of the world, such as Romanian. Romanian belongs to the Romance language group, having many substantial differences from English, including the alphabet, grammar, phonology, etc. In general, the same sentiment is expressed in a more verbose way in Romanian than in English. Ciobotaru and Dinu [17] performed emotion detection over a dataset of about 4000 tweets in Romanian. The texts were manually labeled by them, but the dataset was not made publicly available. Istrati and Ciobotaru [18] collected and manually labeled a dataset with Romanian tweets about brands and created a SA model for usage in brand monitoring and analysis. Unfortunately, their manually labeled dataset is not publicly available. To our knowledge, the recently proposed LaRoSeDa dataset [19] is the first and only public dataset dedicated to sentiment analysis in Romanian. The requirements imposed on our project [20] include a three-class sentiment prediction capability (negative, neutral and positive) and the ability to analyze social media-specific texts. These requirements make LaRoSeDa an unsuitable candidate for our work because the sentiment is labeled in a binary fashion (negative and positive) and the texts refer to product reviews collected from various online shopping platforms, not from social media platforms.

A team at technobium.com (accessed on 26 August 2022) created a commercial engine for SA for the Romanian language, with a free demo posted at sentimetric.ro (accessed on 26 August 2022), capable of determining polarity for various texts, including microblog-ging [21]. However, there are few scientific details on the internal construction of the model and its performance in both prediction capabilities and efficiency with regards to the usage of computational resources. Other efforts worth mentioning in the area of SA for Romanian are [22–26].

To overcome the missing linguistic resources in the underrepresented languages, the works in [27–29] suggest that automated translation could be used for model learning with respect to SA, maintaining a similar performance with the original dataset.

In this research, we restrict our scope to perform SA on Twitter data (as an example of a social network with harsh limit over the size of the messages), on a under-represented language (Romanian) which lacks available labeled data to be used within model training phases. Our final goal is to produce a model for inferring the global polarity of a tweet in a multinomial classification fashion (positive, negative or neutral) with an acceptable performance, even without being in possession of a large Romanian training dataset that can meet our needs. In this respect, we avoid consuming time for a huge data collection and annotation task, and instead, we used an English Twitter dataset of reasonable size translated to Romanian using a public web translation engine.

Our research fits under the wider coverage of a media surveillance project [20], aiming at investigating specific habits of the Romanian public interacting with TV and social media. This imposes several restrictions over our model learning task, such as: the need to re-train the models on a regular basis as the observed environments are being very volatile, the need to process huge loads of messages in a short period of time during audience peaks, and the need to comply with strict data privacy and security standards.

The main contribution of this paper is to demonstrate that, even with the lack of a wellprepared training dataset, good SA results could be obtained by carefully implementing and adapting the standard NLP pipeline for SA [2] to the specificity of the input data. In particular, we show how each step of the NLP pipeline was applied and discuss the consequence of each decision made throughout our NLP experiment. Furthermore, we extend the previous result of Balahur and Turchi [27] obtained using French, German and Spanish tweets to Romanian, showing that comparable performance could be achieved on the translated dataset with that obtained on the original English source.

The paper is organized as follows: in Section 2, we present related work competing with or influencing our research. Section 3 introduces the data under study and the methodological steps followed to construct the SA model. Section 4 presents the obtained results and Section 5 concludes the paper.

#### 2. Related Work

SA is a type of text classification employed with the specific objective of inferring affective states and subjective information from text. To perform SA one could adopt either a specific lexicon-based strategy or could use the standard ML-based text classification pipeline [30]. As indicated by the literature [2,3,31], classical machine learning algorithms or novel deep learning approaches were applied for inferring SA models on texts. Among classical ML algorithms, popular choices [30] are Bernoulli Naive Bayes (NB) [32], Support Vector Machines (SVM) [33], Random Forest (RF) [34] or the Logistic

Regression [35]. For deep learning, all important variants like the standard Deep Neural Network (DNN) [36], the Convolutional Neural Network (CNN) [37], or the Long Short-Term Memory (LSTM) [38] are reported to perform well on text classification tasks. Classical ML methods and the standard DNNs are applied on document-level embeddings such as the well-known TF-IDF [39] or the modern Doc2Vec [40]. Novel methods of DL, including here architectures composed with CNN and LSTM cells are in general applied on word embeddings such as Word2Vec [41]. The classical TF-IDF lead to high-dimensionality prediction problems, thus, the literature [3] suggests considering dimensionality reduction schemes, such as principal component analysis (PCA) [42], non-negative matrix factorization (NMF) [43] or Latent semantic analysis (LSA) [44], to make the computation more efficient. In our work, we will experiment with these methods, in the search for a suitable combination to fit our needs.

Recently, Google proposes BERT [45] as a state-of-the-art pre-trained model for many NLP tasks. Multilingual BERT, pre-trained also for the Romanian language, is reported to be surprisingly good for cross-lingual model transfer [46]. However, as practice indicates [47], BERT comes with significant time costs for model learning—even on very powerful servers, hence this being in contradiction with the need of our project to frequently retrain the models and to accommodate high loads of messages on short time frames. Just for comparison, we will train a BERT-based classifier using its multilingual version for the Romanian language, to see how well we compare with the state of the art, in both classification performance and training time.

Performing SA on microblogging content is seen as a difficult task [12] because one has to deal with bad language [9]. However, for popular languages in the world like English, Spanish or French, plenty of linguistic resources exist such that to enhance SA for microblogging content. For English, we mention BERTweet [10], a large-scale language model trained over a corpus of 850M Tweets, which could be used together with fairseq [13] or transformers [14] for inferring polarity. BERTweet scores an accuracy of 72% on the SemEval2017-Task4A [48] test set, outperforming its competitors RoBERT and XLM-R. Barbieri et al. [49] reports BERTweet as being the State-of-the-Art on TweetEval benchmark, with a 73% average recall. In France, labeled Twitter data were made available for the participants in the DEFT challanges [15], while in Spain, the TASS workshops collected the efforts to classify Spanish annotated tweets [16]. Pota et al. [50] applied BERT-based models for SA on both English and Italian Twitter datasets and drew conclusions about the importance of individualized preprocessing for exploiting hidden information, a suggestion which we specifically followed in our work.

For the Persian language, which is also under-resourced like Romanian, in [51] the lack of any public dataset for sentiment analysis is noted. The authors of this work created a public dataset by collecting 11500 texts and manually labeling each one as either positive, negative or neutral. Around 80% of the texts were collected from an electronic product website, while the rest were collected from Twitter. They applied various classification algorithms and a proposed CNN-LSTM network achieved the highest accuracy of  $\approx$ 85%. In [52], this pretrained hybrid model was used to infer the sentiment polarity of over 800,000 Persian tweets collected over a span of 6 months. The tweets refer to an Iranian COVID-19 vaccine called COVIran Barekat and foreign vaccines (AstraZeneca, Pfizer, Moderna and Sinopharm). The authors compared the sentiments expressed towards the Iranian vaccine versus the sentiments expressed towards the foreign vaccines and found a slight preference for the Iranian one. By obtaining a monthly distribution of opinions, between April and September of 2021, the authors discovered an increase in negative sentiments towards all vaccines between late August and September. A possible explanation could be related to the reported side effects of some of the vaccines. These results seem very promising but, as stated in Section 1, we do not have the necessary resources in order to manually collect and label a large volume of Romanian texts.

The state of the art in SA for microblogging content is less advanced in under-resourced languages such as Romanian. We mention here the efforts of Ciobotaru and Dinu [17] for

emotion detection and Istrati and Ciobotaru [18] for binary sentiment analysis who report promising results, or the company technobium.com (accessed on 26 August 2022) who show a free demo posted at sentimetric.ro (accessed on 26 August 2022). Yet we could hardly rely on those tools to incorporate them in a larger project, as, to our knowledge, a Romanian microblogging content dataset similar to BERTweet is not available.

As stated in Section 1, LaRoSeDa (Large Romanian Sentiment Dataset) seems to be the only publicly available Romanian dataset labeled for sentiment analysis. The dataset contains 15,000 reviews, of which 7500 are labeled positive and 7500 negative. Due to its nature all works using this resource report the performances achieved for sentiment classification in a binary fashion. In [53], an F1 score of 54% is achieved while in the work which introduces the LaRoSeDa dataset an accuracy of  $\approx$ 91% is reported as the benchmark [19]. More recently, we acknowledge the Romanian DistilBERT corpus (https: //github.com/racai-ai/Romanian-DistilBERT, accessed on 30 August 2022) [54] which could be employed for binary SA over standard text data. They report the state-of-the-art binary classification accuracy of 98% for SA performed on LoReSaDa. All the previous experiments were performed on standard text in a binary classification fashion and the problem setting is different from our global polarity inference. Regarding the multinomial SA of social media texts in Romanian, we could not find any published work in order to set a benchmark with which we can compare.

Searching for other Romanian research relevant to SA, we mention Lupea and Briciu [22] who developed a Romanian Emotions Lexicon, attaching tags to words. Tufis and Barbu Mititelu [26] developed RoWordNet (https://github.com/dumitrescustefan/RoWordNet, accessed on 30 August 2022), a semantic network of words based on the idea introduced by the WordNet English lexicon—each word having attached a polarity score. Both lexicons could be of use in sentiment analysis if a dictionary-based method (as defined in [3] p. 541) is selected for SA. Other Romanian authors performed sentiment analysis on different sort of input data than tweets, such as speech [23,24] or poetry [25], thus benefiting from resources developed with other purposes, such as SRoL [55] which was developed to help Romanian speech processing research.

Constructing language resources for under-represented languages by automatic translation seems to work well for various NLP tasks, as acknowledged in [27–29,56]. Balahur and Turchi [27,29] shows that automatic translation of datasets performed with Google Translate, Bing Translator and Moses works well for French, Spanish and German in relation to the SVM SMO classifier. Balahur and Perea-Ortega [28] performed extensive experiments with English and Spanish Twitter datasets supplied for SemEval 2013 and TASS 2013 workshops and showed that training data obtained from machine translated text could work well for learning polarity classification systems. Banea et al. [56] positively respond to the question whether we can "reliably predict sentence-level subjectivity in languages other than English by leveraging on a manually annotated English dataset" by learning Naive Bayes classifiers on six languages, including Romanian, starting from an original English dataset with news articles translated with automatic engines available at that moment in time. Thus, this further motivates our efforts to use machine translation for obtaining the learning dataset for Romanian.

#### 3. Data Processing Methodology

Given the specific advice provided in [2] with respect to building a SA classifier, in this section, we present our approach to carrying out this task. Figure 1 summarizes all performed steps. At the top of the diagram, the automatic translation process of the dataset from English to Romanian is highlighted. The preprocessing step consists of various procedures and are grouped in two different abstract pipelines. The one on the left will generate texts which are fit for the TF-IDF variants, Doc2vec and Word2Vec techniques while the one on the right will generate texts as expected by the pretrained BERT encoder. The feature extraction step highlights the transformation of the preprocessed text into numeric features which can be later used to train the ML models. The TF-IDF variants and Doc2Vec are grouped together to highlight that the output of all these methods are in the same form. To be more specific, a preprocessed text is transformed into a vector of length N while Word2Vec will transform a preprocessed text into a NxM matrix where the number of rows will be equal to the number of words within the text and the number of columns will be equal to the word embedding size, i.e., the number of values set to represent a single word/token. BERT contextual encoding will represent texts using multiple vectors.



Figure 1. Architecture of the sentiment analysis system.

In model training and tuning, we can see that for all the selected ML approaches, with the exception of the BERT classifier, the evolutionary hyperparameter optimization methodology is used to identify the optimal set of parameters. Due to the high training times of BERT, we opted for a classic fine tuning process using the recommended parameters. The Bernoulli NB, Linear SVM, Logistic Regression, Random Forest and DNN are grouped together to highlight the many-to-many relation of this group with the TF-IDF variants and Doc2Vec features. This means that any feature from this group can be used by any model mentioned previously. LSTM and CNN are grouped together in order to highlight that both use the Word2Vec features, while the BERT classifier uses the specific BERT encodings. The prediction of any trained model will denote the sentiment polarity of the input texts.

In the following subsections, we introduce the dataset and provide details on the following: text preprocessing, feature extraction, dimensionality reduction, and classifier selection. Many transformations over the text indicated below were implemented with the help of SpaCy [57] (https://spacy.io/, accessed on 1 September 2022). Model construction and performance evaluation are presented in Section 4.

#### 3.1. Dataset

For our research, we selected the Twitter US Airline Sentiment Tweets (https://www. kaggle.com/crowdflower/twitter-airline-sentiment, accessed on 1 September 2022) dataset. The data were collected in 2015 and each tweet was manually labeled by external contributors with its global polarity (positive, negative and neutral). It contains around 15,000 tweets, 63% being negative, 21% neutral and 16% positive. Each tweet is accompanied by the contributor's confidence about the annotated sentiment and each negative tweet is accompanied by a reason for the assessment. There are a number of reasons why we consider this dataset suitable for our research purpose: (i) it contains microblogging-specific (bad) language, (ii) the sentiment class of each tweet was manually annotated and we can easily verify the correctness and the reason for the annotation, (iii) the number of tweets is large enough to provide reasonable training data, and (iv) the tweets are relatively recent.

As a first processing step, we used Google Translate (https://translate.google.com/, accessed on 2 September 2022) service to translate all the tweets of the dataset in Romanian. We eliminated all duplicated rows and sorted the dataset by Tweet Id. Thus, we obtained two datasets: the original Twitter US Airline Sentiment Tweets (in English) and its Romanian translation.

The structural, grammatical and syntactical integrity of any text translated with automated processes is affected. The main metric used in the literature to measure the quality of an automated translator is the BLEU (Bilingual Evaluation Understudy) score. This score is computed by comparing a translation with one or more acceptable translations and checking for the presence/absence of particular words, the word ordering and the degree of distortion. If the score measures from 0 to 100, a higher number represents a better translation (100 denoting perfect translation). In [58], general English texts are translated to 50 different languages using Google Translate and the BLEU score is computed. The mean score over all the compared translations is  $\approx$ 76. English to Romanian achieved a score of 84, which is considerably above average. The maximum BLEU score of 91 was achieved by English to Portuguese while the minimum of 55 was achieved by English to Hindi. Similar results are also obtained in [59] where English to Romanian obtained better than average results. In both works, better translations are obtained for languages which are in the same of similar family with English. Translations from English to distant languages, such as Hindi or Hebrew, are the most negatively effected.

For the purpose of running ML tasks, the dataset was split in training and test sets. The training dataset consists of approx. 11,000 instances, while the testing dataset consists of the remaining approx. 3700 instances, thus ensuring a 75–25% split between train and test data. The split was made just after text preprocessing, such that the class distribution between the train and test set was similar. Moreover, the English and Romanian train and test set are identical in the sense that they contain the same instances.

#### 3.2. Text Preprocessing

Dealing with bad language is mandatory in a SA task over microblogging content [10], thus, in this subsection we describe specific text preprocessing efforts done with this respect. As indicated by Pota et al. [50], individualized pre-processing of the tweets is required in order to better exploit the hidden information of the input data. We developed a specialized preprocessing module, containing the following steps, applied in this specific order:

- 1. Extra white space removal (language-independent);
- 2. Custom word lemmatization and tokenization (language-dependent);
- 3. URL identification and removal (language-independent);
- 4. Emoji identification and replacement (language-independent);
- 5. Social media mention identification and removal (language-independent);
- 6. Extra consecutive character removal (language-independent);
- 7. Abbreviation replacement (language-dependent);
- 8. Stop-word removal (language-dependent);
- 9. Lower case capitalization (language-independent);
- 10. Punctuation mark removal (language-independent).

Language-independent steps can be applied in the same manner in both English and Romanian. In contrast, language-dependent steps implies that specific knowledge of Romanian or English is requested.

For building the BERT-based classifier, we performed only steps 3 and 5, and we performed an additional sentence-level tokenization. Next, we called to the BertTokenizer to obtain the specific BERT encodings.

All steps, excluding 2 and 4, are commonly used for text preprocessing and were applied in our work in the recommended fashion. In step 2, we instructed Spacy [57] to not lemmatize social media-specific tagged words, to not split the tagged word during the tokenization phase, and to not remove the negation element during the stop-word removal.

Step 4 is critical for our microblogging context, as we deal with emojis, as a sort of Twitter bad language. Kralj Novak et al. [60] reports that about 4% of Tweets contain emojis and their sentiment polarity does not depend on the language. In this respect, they constructed the Emoji Sentiment Ranking lexicon containing the 751 most frequently used emojis, each annotated with the sentiment polarity (negative, neutral or positive). Here, we verify whether a token is found in the Full Emoji List (https://unicode.org/emoji/charts/full-emoji-list.html, accessed on 3 September 2022) and whether it has an associated sentiment in the Emoji Sentiment Ranking lexicon mentioned above. If true, the token will be replaced with its polarity and a special prefix and suffix.

#### 3.3. Feature Extraction

Key to any NLP task is the document internal representation, i.e., properly selecting the features from the raw text and encoding them to numerical values, so as to keep the representation tractable or to enrich it with some language semantics [3]. In our work, we tested the most popular approaches as suggested by the NLP literature [2,3]: TF-IDF, Word2Vec and Doc2Vec. We learned the embeddings on the English dataset and its Romanian translation and we restricted the vocabulary to contain only the tokens which appear at least three times, removing a large number of infrequent tokens or those which may have been erroneously built in the preprocessing step.

We notice that the resulting English vocabulary contains around 3100 tokens, while the Romanian one contains around 4000, due to the fact that Romanian is more verbose than English.

In order to learn the Word2Vec and Doc2Vec embeddings for our data, we used the Gensim library [61]. For Word2Vec, we worked with the Continuous Bag-Of-Words (CBOW) architectural model. For learning the Doc2Vec embedding, we used the Distributed Bag-Of-Words (DBOW) with hierarchical softmax architecture. In both cases, we set the vector embedding size for each token to 200 and we trained each model with the following parameters: learning rate  $\alpha = 0.025$ , *window* = 5, over 5 *epochs*.

For all three embeddings, the trained models were then applied on the testing sets.

### 3.4. Dimensionality Reduction

The abovementioned document representations and especially TF-IDF lead to highdimensionality prediction problems [3], causing learning algorithms to run slowly or requesting huge memory resources [31].

Therefore, we applied PCA [42], NMF [43] and LSA [44] dimensionality reduction algorithms on the datasets represented with TF-IDF features, reducing the number of features to 500. This means that the reduced representation for English is around 6.2 times smaller and for Romanian—about 8 times smaller.

We did not apply dimensionality reduction on the Word2Vec and Doc2Vec data because the desired vector size of each representation is set before the feature extraction, in our case: 200.

#### 3.5. Classifier Selection

Following suggestions in the literature [2,3,31], we selected the following methods for building our classifiers:

- Bernoulli Naive Bayes (Bernoulli NB), Support Vector Machine (SVM), Random Forest (RF) and Logistic Regression (LR) from the classical ML;
- 2. Deep Neural Network (DNN), Long Short-Term Memory (LSTM) and the Convolutional Neural Network (CNN) from the area of deep learning;
- 3. Multilingual BERT—to get a glimpse of the state-of-the-art results.

As indicated in Figure 1, the classical ML methods and the DNN were applied on the TF-IDF encoding, with and without dimensionality reduction and on Doc2Vec. On Word2Vec, we constructed classifiers with the help of LSTM and CNN.

We implemented the classical ML algorithms with the help of Scikit-Learn library [62], while for the deep learning we used Keras [63].

For BERT, we used the model available on the Hugging Face transformers (https: //huggingface.co/docs/transformers/model\_doc/bert, accessed on 5 September 2022), called with the base multilingual uncased variant. On top of BERT, we added a hidden dense layer with 75 nodes and ReLU activation function, followed by the standard classification layer with 3 nodes which produce the sentiment. Adam was the selected optimizer, with a learning rate of  $2 \times 10^{-5}$  and  $\epsilon = 10^{-8}$ . The loss function was set to Categorical CrossEntropy.

#### 3.6. Hyperparameter Optimization

When applying each of the abovementioned learning algorithms, we need to tune them with proper parameters selected so as to minimize the generalization error [64,65]. Various approaches could be considered such as exhaustive grid search, random search [65], Bayesian optimization [66] or evolutionary optimization (EO) [67].

Given the large number of parameters to optimize, and noticing the vast literature accompanying the metaheuristic design of DNNs [36] or recent applications of DL where parameters were selected with the help of genetic algorithms [68–70], or suggestions that EO could outperform Bayesian optimization [71], we decided to employ a classical genetic algorithm for hyperparameter search.

We used Sklearn-genetic-opt library [72] for implementing genetic algorithm-based hyperparameter optimization in relation with our selected algorithms. Sklearn-genetic-opt makes usage of the Deap framework (https://github.com/deap/deap, accessed on 5 September 2022) [73], which supplies many evolutionary algorithms needed for solving optimization problems.

The GA was designed as following. Given a number *N* of parameters to optimize for some specific learning method, a chromosome is a vector  $(p_1, p_2, ..., p_N)$  of values

selected for each parameter. A population consists of 20 individuals which is evolved over 40 generations with a crossover probability of 0.8 and mutation probability of 0.1. Individuals are selected for the next generation with a standard elitist tournament of size 3. Internally, each individual is evaluated using the accuracy as fitness function, computed with 3-fold cross-validation.

In the case of DNN, we considered among the parameters the following: the network capacity (the number of hidden layers and the number of units per layer), the activation function, the regularization function, the drop-out rate

Since both CNN and LSTM need the embedding weight parameter which is 2D tensor, we modified the source code of Sklearn-genetic-opt, in order to transmit the multidimensional parameters directly to Deap.

The Appendix A presents in full the parameters considered for evolutionary optimization, for all classifiers.

In general, convergence is seen after 15–20 generations, thus evolving the populations over 40 generations is more than enough to guarantee a good parameter selection.

For the BERT-based classifier, as learning just one model is very time consuming, we omitted to perform the evolutionary optimization procedure. Instead of cross-validation, we took 10% of the training set for validation and we let the learning to optimize the loss function for several epochs. We noticed that the model rapidly overfits, thus, we stop the learning after two epochs.

#### 4. Experiments and Results

In this section, we present our experiments and discuss the results. We first construct models on both the original and translated Twitter US Airline Sentiment Tweets dataset and next, we investigate how the best obtained Romanian models models perform on small real-life Romanian datasets, manually labeled.

#### 4.1. Constructing The Models

As mentioned in Section 3, we applied the processing pipeline described in Figure 1 on the Twitter US Airline Sentiment Tweets (https://www.kaggle.com/crowdflower/twitter-airline-sentiment, accessed on 1 September 2022) dataset, which we translated in Romanian with a public web translation service. Sections 3.2–3.6 present all the details regarding every processing step of the pipeline.

All the experiments were conducted on a powerful machine with the following specifications:  $2 \times$  Intel Xeon Gold 6230 CPUs (20 Core at 2.1 GHz), 128 GB DDR4 internal RAM,  $8 \times$  NVIDIA Tesla V100 32GB and the source code was implemented in Python 3.9.

Table 1 presents the learning performance on the classifiers mentioned in Section 3.5 on the test set, with or without dimensionality reduction. We assess the classification performance with the help of the accuracy and the weighted F1-measure. We also report the performance obtained on the original English data set, applying the same data processing pipeline (without any dimensionality reduction), in order to see how much we lose by the automatic translation to Romanian.

We observe that the classification performance of all considered models trained on the original English dataset is very close to the one obtained on the Romanian translation. The differences in all classification schemes are  $\pm 1.5\%$  which can be considered negligible. We expected this result, as it is in line with similar experiments done with automatic translation for other languages [27–29]. Furthermore, they confirm the validity of the processing and learning pipeline, as applied on the Romanian translated Twitter data.

We note that dimensionality reduction does not bring in an increase in classification accuracy. Furthermore, similar accuracies of around 78% are obtained either with TF-IDF or with Word2Vec feature extraction, but Doc2Vec does not help, in any scenario.

In terms of accuracy, Bernoulli NB, SVM, LR and DNN applied on TF-IDF encoding and CNN and LTSM applied on Word2Vec are all almost similar. However, Bernoulli NB scores slightly better on weighted F1-measure, therefore, we are prompted to select Bernoulli NB as the best classifier for TF-IDF encoding. For Word2Vec encoding, LSTM slightly outperforms CNN in both accuracy and weighted F1-measure.

**Table 1.** Classifier performance (accuracy and weighted F1-measure) with or without dimensionality reduction.

Encoding	Classifier	English Original Datasat	Romanian Translated Dataset				
Lincouning	Classifier	English Onginal Dataset	Only PP	PP + PCA	PP + NMF	PP + LSA	
	Bernoulli NB	0.7737, 0.7725	0.7820, 0.7820	0.6778, 0.6249	0.7293, 0.7222	0.6701, 0.6077	
	SVM	0.7741, 0.7667	0.7836, 0.7747	0.7620, 0.7471	0.7316, 0.6988	0.7690, 0.7565	
TF-IDF	RF	0.6670, 0.5521	0.6520, 0.5471	0.6366, 0.5716	0.6541, 0.5517	0.6410, 0.5787	
	LR	0.7745, 0.7590	0.7781, 0.7645	0.7573, 0.7347	0.7499, 0.7388	0.7600, 0.7430	
	DNN	0.7818, 0.7715	0.7720, 0.7623	0.7658, 0.7575	0.7502, 0.7307	0.7636, 0.7507	
Ward2Vac	CNN	0.7821, 0.7666	0.7769, 0.7600				
word2vec	LTSM	0.7750, 0.7635	0.7817, 0.7798				
	Bernoulli NB	0.6244, 0.4801	0.6242, 0.4798				
	SVM	0.6305, 0.4801	0.6267, 0.4797				
Doc2Vec	RF	0.6244, 0.4790	0.6242, 0.4779				
	LR	0.6275, 0.4800	0.6252, 0.4797				
	DNN	0.6290, 0.4801	0.6273, 0.4798				
Multilingual BERT		0.8302, 0.8257	0.8099, 0.8051				

We took advantage of a very powerful machine to run all the experiments. Even so, time spent for hyperparameter optimization and model learning are not negligible. Table 2 lists the time spent for Evolutionary optimization and for learning the final model with the optimal parameter set, for each tested classifier. EO helped us to achieve a 1–3% improvement for the weighted F1-measure.

Table 2. Hyperparameter optimization and final model learning time (seconds).

Encodino	Classifian	English Original Dataset	Romanian Translated Dataset				
Encouring	Classifier	English Original Dataset	Only PP	PP + PCA	PP + NMF	PP + LSA	
	Bernoulli NB	1337, 0.285	1645, 0.368	387, 0.064	362, 0.051	385, 0.062	
	SVM	920, 0.360	1048, 0.238	2038, 2.418	412, 0.149	2724, 2.353	
TF-IDF	RF	5735, 1.590	6158, 0.568	375, 0.295	449, 0.404	659, 0.905	
	LR	89,603, 8.502	3588, 7.855	2245, 4.792	5125, 16.644	204, 0.693	
	DNN	11,513, 2.176	13,551, 2.230	3558, 1.015	9236, 5.280	6585, 2.567	
MandQVaa	CNN	9143, 1.460	16,127, 4.209				
woru2 vec	LTSM	17,172, 16.865	62,364, 20.926				
	Bernoulli NB	271, 0.028	274, 0.021				
	SVM	654, 0.190	580, 0.152				
Doc2Vec	RF	712, 1.171	263, 0.523				
	LR	484, 1.595	428, 0.936				
	DNN	17,842, 0.912	11,529, 4.594				
Multilingual BERT <sup>a</sup>		416	444				

<sup>a</sup> Time needed for learning a single BERT-based model.

In Table 2, we can also observe that in the majority of cases, the hyperparameter optimization process and the final model training times are higher for the Romanian classifiers. As stated in Section 3.3, Romanian is more verbose than English, thus, the number of tokens learned for Romanian is larger. This fact might have contributed to the generation of more complex models when compared to English.

We shall note that, in general, learning a classical ML model takes less than a second and this is clearly less than the learning time required for a deep learning model, but searching for the best model parameters indeed is very time consuming. In the case of the models trained in Romanian, optimizing the Bernoulli NB on the dataset without dimensionality reduction took about 27 min. Searching for the best structure and capacity of the DNN took about 3 h and 45 min. Searching for best parameters for LTSM took more than 17 h, and we shall note that vector embedding size is only 200. Therefore, learning an LSTM proved to be a very prohibitive experiment in the absence of a well-equipped computing machine.

As expected, BERT provides with state-of-the-art results for both the English and the Romanian datasets. The gap from our best result to BERT is bigger for English compared with Romanian. However, all come with a high computational cost, as learning just one BERT-based classifier takes about 7 to 8 min. This makes the hyperparameter optimization infeasible. If we would consider the specification of the Genetic Algorithm presented in Section 3.6, this would result in the worst case at 16,800 models learned, needing about 11 days of running the experiment. However, as we noticed, we do not need to perform this optimization, as just one BERT-based model learned with the recommended parameters already supplies state-of-the-art results. The problem with a BERT-based classifier is not with learning one model, but with time needed to classify unknown instances. Whereas, the other classifiers took negligible time (less than 1 s) to process the test set, the BERT-based model took about 44 s. This would prohibit us from employing the BERT-based classifier in the media surveillance situations with an extreme high-throughput of messages (e.g., during a prime-time audience TV show).

Given the difference of only 2% between the best achieved performance (Bernoulli NB and the LSTM classifiers) and BERT, taking also into account the reasonable learning and testing time of those models, we conclude that we could strongly consider the classical Bernoulli NB as being our choice for the production environment required by our project.

#### 4.2. Assessing the Models Performance on Real Cases

Given that the final purpose in our project is to apply the learned models for inferring the polarity of any Romanian tweet, we manually labeled two small test sets, each one containing 120 distinct tweets. The first one includes tweets specific to the airline industry, comparable with the ones used for training our models, and the second one includes general tweets. We applied on them the best models reported in the previous subsection (i.e., Bernoulli NB for TF-IDF encoding and LSTM for Word2Vec), the public demo of sentimetric.ro (accessed on 8 September 2022) [21] and the BERT-based classifier.

Each tweet was manually labeled by five human volunteers. Each one expressed an opinion about the polarity of the tweet and the final sentiment was established to be the one that was selected by the majority. Labeling statistics regarding how humans assessed the polarity is presented in Table 3. We shall note that the labeling task seemed to be a difficult one for the volunteers, as for only 43 tweets (35.8%) in the case of airline industry specific dataset and 47 tweets (39.2%) in the case of general tweets all the 5 volunteers reached a unanimous decision. Furthermore, the distribution of the polarity of the tweets significantly differ from the one of the Twitter US Airline Sentiment Tweets (presented in the last row of Table 3).

**Table 3.** Labeling statistics regarding how humans manually assessed the polarity (number of tweets and percentage).

Dataset	Negative	Neutral	Positive	Unanimous Annotation
Airline industry-specific tweets	51, 0.465	36, 0.300	33, 0.275	43, 0.358
General tweets	45 <i>,</i> 0.375	32, 0.266	43, 0.358	47, 0.392
Twitter US Airline Sentiment Tweets	0.63	0.21	0.16	

Polarity estimation results on the Romanian dataset with airline industry-specific tweets are presented in Table 4.

Both encodings supply better results with our models (Bernoulli NB and LSTM) than sentimetric.ro. This is expected as we learned our models on tweets specific for the aviation domain. BERT outperforms Bernoulli NB only by a slight margin.

**Table 4.** Performance of various models (accuracy and weighted F1-measure) on a Romanian dataset with aviation industry-specific tweets.

Encoding	Classifier	Accuracy	Weighted F1-Measure
TF-IDF	Bernoulli NB	0.6500	0.6311
Word2Vec	LSTM	0.5833	0.5518
Sentimetric.ro		0.4750	0.4699
Multilingual BE	RT	0.6583	0.6338

Table 5 presents the models results on the Romanian general tweets dataset. We notice that Bernoulli NB scores better than sentimetric.ro (accessed on 8 September 2022) in terms of the weighted F1-measure. LSTM scores worse. sentimetric.ro (accessed on 8 September 2022) proved to assess a better polarity on the general domain than on the aviation, which is expected, as we assume that the engine was constructed for a wide usage. BERT proves to be the state of the art, as the margin by which it outperforms our best models is around 5%.

For both domains, our models' results are worse than those obtained on the translated test set used in Section 4.1, because now the tweets are real ones, not translated, and their target class distribution differs significantly—i.e., from a statistical point of view, sets are extracted from different statistical populations.

Bernoulli NB is more robust to novel tweets and to a different domains than LSTM and we suppose that this happens because for LSTM the Word2Vec embedding is learned on our very limited translated dataset and not on the whole Romanian language.

**Table 5.** Performance of various models (accuracy and weighted F1-measure) on a Romanian dataset with general tweets.

Encoding	Classifier	Accuracy	Weighted F1-Measure
TF-IDF	Bernoulli NB	0.4833	0.4942
Word2Vec	LSTM	0.4583	0.4429
Sentimetric.ro		0.4917	0.4730
Multilingual BE	RT	0.5583	0.5417

#### 4.3. Discussion and Further Work

Experiments presented in Section 4.1 show that a standard method such as Bernoulli Naive Bayes employed on the classical TD-IDF encoding supplies results that fit our media surveillance needs. The performance of the Bernoulli NB classifier is slightly better than the ones of other classifiers, being in a narrow margin below a BERT-based classifier. Applying evolutionary optimization for the hyperparameter search allows us to improve the performance of all classifiers by 1% to 3%.

Bernoulli NB has the advantage of very fast inference times for novel instances, being also easily retrainable, to accommodate for the volatility of the discussed topics. In contrast, although the BERT-based classifier indeed produces the state-of-the-art results in terms for both accuracy and weighted F1-measure, its needs in terms of hardware resources and computational time make it infeasible for our practical needs.

With the final experiments presented in Section 4.2, we demonstrate that the selected classification model is suitable for our project production environment with general discussion topics, with a performance superior or at least equivalent to the already existing classifiers for the Romanian language although: (i) we learned a limited language model

from a very specific dataset of only about 15,000 tweets; (ii) Romanian knowledge was produced with a public automatic translation service; and (iii) model learning was performed on domain-specific knowledge—the airline industry.

Therefore, we confirm that using datasets for Romanian NLP tasks constructed by automatic translation from English could be a solution, especially for SA. If no extensive language model exists, and if computing time and hardware resources represent a barrier, we recommend the use of the very classical TF-IDF encoding with a simple classifier such as the Bernoulli NB. In our case, classical ML methods proved to be more robust to generalization than DL-based methods. This might be due to the fact that the classic models used the TF-IDF features which do not take into account semantic and syntactic structures, thus being less affected by the automatic translation process when compared to Word2Vec. If one would opt for a wide-scope hyperparameter search, we suggest to employ evolutionary optimization as being capable of fine tuning the classifiers in a reasonable time.

For our future work, we would like to repeat the experiment presented in Section 4.2 with the following differences: using considerably more texts in order to be more representative, defining standardized labeling rules, employing the help of more human annotators, and excluding texts which proved to be hard to label even by humans. Having more texts which are better labeled by more human annotators should improve the accuracy of all the presented models. Additionally, the sentiment class distribution of the dataset could be balanced using various oversampling techniques. The models can be retrained on the balanced data following the same methodology presented in this work and their performance re-evaluated. If these modifications do not bring a significant improvement, the LIME (https://github.com/marcotcr/lime, accessed on 10 September 2022) (Local Interpretable Model-Agnostic Explanations) model can be used to understand the reasons behind predictions and decide which models are more robust and trustworthy [74].

#### 5. Conclusions

Within the larger setup of a media surveillance project [20], we constructed a system capable of inferring the global sentiment polarity of Romanian tweets starting from an English dataset specific to the aviation industry, translated to Romanian. This paper describes our experience in designing the classification system and extracts several noteworthy conclusions in sentiment analysis of microblogging content for Romanian. As similar works treat the Romanian SA task only as a binary classification, we set the benchmark accuracy for a multinomial methodology consisting of three classes: negative, positive, and neutral. Bernoulli NB trained on TF-IDF features achieved an accuracy of around 78%, while BERT achieved the best result of 81%.

After carefully processing the Twitter data in order to properly approach bad language in particular, we built and evaluated models constructed with the help of various classifiers, including standard machine learning or the very popular nowadays deep learning. Given the large number of parameters to optimize for fine-tuning the classifiers, we opted to perform hyperparameter search with the help of evolutionary optimization. We found that the Bernoulli Naive Bayes classifier is the most robust one to both aviation industry specific tweets or to general ones and TF-IDF encoding should be used if no additional linguistic resources are available.

Regarding the performance measured with the help of accuracy and weighted F1measure, we notice that it does not differ significantly for the English original dataset and its Romanian translation. Furthermore, although the training data were specific to aviation industry, classification performance achieved on a small dataset with general tweets seems to be slightly better than one of a commercial public demo available on the market.

Learning standard deep neural networks on a TF-IDF encoding or LSTM on a Word2Vec encoding bring in comparable results, but with an increased computational cost. Doc2Vec encoding seems not to help, as results are worse, regardless of the classifier.

Further research is still needed in order to obtain even better results. LSTM requires pre-trained Word2Vec embeddings for the target language, which are not available for Romanian. Moreover, a larger and more balanced dataset on the general domain could be used for learning, but probably with an increased computational cost.

Author Contributions: Conceptualization, D.C.N., A.B.R., M.G. and M.A.B.; methodology, D.C.N., A.B.R., M.G. and M.A.B.; software, D.C.N., M.G., M.A.B., N.B. and A.G.; validation, D.C.N., A.B.R., M.G., M.A.B., N.B. and A.G.; tormal analysis, D.C.N.; investigation, D.C.N., M.G. and M.A.B.; resources, A.B.R.; data curation, D.C.N. and M.G.; writing—original draft preparation, D.C.N. and A.B.R.; writing—review and editing, D.C.N., A.B.R., M.G., M.A.B., N.B. and A.G.; visualization, D.C.N. and A.B.R.; broject administration, A.B.R.; funding acquisition, A.B.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This paper was financed by the project with the title "Platformă inovativă pentru măsurarea audienței TV, identificarea automată a telespectatorilor și corelarea cu date analitice din platforme de socializare online" (Innovative platform for measuring TV audience, automatic identification of viewers and correlating it with analytic data from social media). The project was co-financed by "Fondul European de Dezvoltare Regională prin Programul Operațional Competitivitate (POC) 2014–2020, Axa prioritară: 2-Tehnologia Informației și Comunicațiilor (TIC) pentru o economie digitală competitivă." (the European Regional Development Fund (ERDF) through the Competitiveness Operational Program 2014-2020, Priority Axis 2—Information and Communication Technology (ICT) for a competitive digital economy), project code SMIS 2014+: 128960, beneficiary: CICADA TECHNOLOGIES S.R.L. The project is part of the call: POC/524/2/2/ "Sprijinirea creșterii valorii adăugate generate de sectorul TIC și a inovării în domeniu prin dezvoltarea de clustere" (Supporting the added value generated by the ICT sector and innovation in the field through cluster development). The content of this material does not necessarily represent the official position of the European Union or the Romanian Government.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

**Data Availability Statement:** Publicly accessible data from Kaggle were used https://www.kaggle.com/crowdflower/twitter-airline-sentiment (accessed on 10 September 2022).

Conflicts of Interest: The authors declare no conflict of interest.

#### Abbreviations

The following abbreviations are used in this manuscript:

SA	Sentiment Analysis
NLP	Natural Language Processing
ML	Machine Learning
NB	Naive Bayes
SVM	Support Vector Machine
RF	Random Forest
LR	Logistic Regression
DNN	Deep Neural Network
LSTM	Long Short-Term memory
CNN	Convolutional Neural Network
BERT	Bidirectional Encoder Representations from Transformers
TF-IDF	Term Frequency-Inverse Document Frequency
GA	Genetic Algorithm
EO	Evolutionary Optimization
PCA	Principal Component Analysis
NMF	Non-negative Matrix Factorization
TCA	

LSA Latent Semantic Analysis

# Appendix A

**Table A1.** Parameters and their values and ranges considered for EO hyperparameter search: classical learners.

Classifier	Parameter Name	Space Type	Variable Type	Values or Range
Bernoulli NB	alpha	continuous	float	[0.0, 1.0]
	binarize	continuous	float	[0.0, 1.0]
	fit_prior	categorical	boolean	[True, False]
Linear SVM	dual	categorical	boolean	[False]
	C	continuous	float	[0.0, 1.0]
	penalty	categorical	string	["11", "12"]
	fit_intercept	categorical	boolean	[True, False]
	intercept_scaling	continuous	float	[1.0, 10.0]
	class_weight	categorical	string or null	[None, "balanced"]
	tol	categorical	float	[0.0001]
	loss	categorical	string	["squared_hinge"]
	multiclass	categorical	string	["ovr"]
	max_iter	continuous	int	[200, 2000]
Logistic regression	penalty	categorical	string	["l2", "none"]
	dual	categorical	bolean	[False]
	tol	categorical	float	[0.0001]
	C	continuous	float	[0.1, 1.0]
	fit_intercept	categorical	boolean	[True, False]
	intercept_scaling	continuous	float	[1.0, 10.0]
	class_weight	categorical	string or null	[None, "balanced"]
	solver	categorical	string	["newton-cg", "lbfgs", "sag", "saga"]
	max_iter	continuous	int	[100, 1500]
	multiclass	categorical	string	["multinomial"]
	warm_start	categorical	boolean	[True, False]
Random Forest	n_estimators	continuous	int	[20, 200]
	criterion	categorical	string	["gini", "entropy"]
	max_depth	categorical	null	[None]
	min_sample_split	continuous	int	[2, 20]
	min_sample_leaf	continuous	int	[1, 10]
	max_features	categorical	string or null	["auto", "sqrt", "log2", None]
	max_leaf_nodes	categorical	null	[None]
	min_impunity_decrease	continuous	float	[0.0, 0.2]
	bootstrap	categorical	boolean	[True]
	oob_score	categorical	boolean	[True, False]
	warm_start	categorical	boolean	[False]
	class_weight	categorical	string or null	["balanced", "balanced_subsample", None]
	ccp_alpha	continuous	float	[0.0, 0.75]

Classifier	Parameter Name	Space Type	Variable Type	Values or Range
DNN parameters	batch_size	continuous	int	[32, 1024]
	epochs	continuous	int	[1, 20]
	activation	categorical	string	["relu", "sigmoid", "softmax", "softplus", "softsign", "tanh", "selu", "elu", "exponential"]
	kernel_initializer	categorical	string	["random_normal", "random_uniform", "truncated_normal", "glorot_normal", "glorot_uniform", "he_normal", "he_uniform", "identity", "orthogonal", "variance_scaling", "lecun_normal", "lecun_uniform", "zeros"]
	optimizer	categorical	string	["adadelta", "adagrad", "adam", "adamax", "ftrl", "nadam", "rmsprop", "sgd"]
	use_bias	categorical	boolean	[True]
	bias_initializer	categorical	string	["random_normal", "random_uniform", "truncated_normal", "glorot_normal", "glorot_uniform", "he_normal", "he_uniform", "variance scaling", "lecun normal", "lecun uniform", "zeros"]
	kernel regularizer	categorical	string or null	[None, "]1", "]2"]
	bias regularizer	categorical	string or null	[None, "11", "12"]
	loss	categorical	string	["sparse categorical crossentropy"]
	dropout_rate	continuous	float	[0.0, 0.7]
DNN structure	n_layers <sup>a</sup>	continuous	int	[1, 10]
	first laver nodes b	continuous	int	[128, 1024]
	last_layer_nodes <sup>c</sup>	continuous	int	[8, 128]

**Table A2.** Parameters and their values and ranges considered for EO hyperparameter search: the case of DNN.

<sup>a</sup> Number of hidden layers. If set to 1, last\_layer\_nodes parameter will be ignored. If greater than 2, then the number of nodes in the middle layers will be computed using a linear scaling function. <sup>b</sup> Number of nodes for the first hidden layer. <sup>c</sup> Number of nodes for the last hidden layer.

**Table A3.** Parameters and their values and ranges considered for EO hyperparameter search: the case of CNN.

Classifier	Parameter Name	Space Type	Variable Type	Values or Range
CNN parameters	batch_size	continuous	int	[32, 1024]
-	epochs	continuous	int	[1, 20]
	activation	categorical	string	["relu", "sigmoid", "softmax", "softplus", "softsign", "tanh", "selu", "elu", "exponential"]
	kernel_initializer	categorical	string	["random_normal", "random_uniform", "truncated_normal", "glo- rot_normal", "glorot_uniform", "he_normal", "he_uniform", "orthog- onal", "variance_scaling", "lecun_normal", "lecun_uniform", "zeros"]
	optimizer	categorical	string	["adadelta", "adagrad", "adam", "adamax", "ftrl", "nadam", "rm- sprop", "sgd"]
	use_bias	categorical	boolean	[True]
	bias_initializer	categorical	string	["random_normal", "random_uniform", "truncated_normal", "glo- rot_normal", "glorot_uniform", "he_normal", "he_uniform", "vari- ance_scaling", "lecun_normal", "lecun_uniform", "zeros"]
	kernel_regularizer	categorical	string or null	[None, "l1", "l2"]
	bias_regularizer	categorical	string or null	[None, "l1", "l2"]
	activity_regularizer	categorical	string or null	[None, "l1", "l2"]
	mask_zero	categorical	boolean	[False, True]
	kernel_size	continuous	int	[1, 5]
	padding	categorical	string	["same", "valid"]
	pool_size	continuous	int	[1, 5]
	pool_strides	continuous	int	[1, 5]
	loss	categorizal	string	["sparse_categorical_crossentropy"]
	dropout_rate	continuous	float	[0.0, 0.7]
CNN structure	n_layers <sup>a</sup>	continuous	int	[1, 10]
	first_layer_filters <sup>b</sup>	continuous	int	[32, 128]
	last_layer_filters <sup>c</sup>	continuous	int	[8, 64]

<sup>a</sup> Number of hidden layers. If set to 1, last\_layer\_filters parameter will be ignored. If greater than 2, then the number of filters of the layers between the first and the last will be computed using a linear scaling function. <sup>b</sup> Number of filters for the first hidden layer. <sup>c</sup> Number of filters for the last hidden layer.

Classifier	Parameter Name	Space Type	Variable Type	Values or Range
LSTM parameters	batch_size epochs activation recurrent activation	continuous continuous categorical categorical	int int string string	[32, 1024] [1, 20] ["tanh"] ["sigmoid"]
	kernel_initializer	categorical	string	["random_normal", "random_uniform", "truncated_normal", "glorot_normal", "glorot_uniform", "he_normal", "he_uniform", "identity", "orthogonal", "variance_scaling", "lecun_normal", "lecun_uniform", "zeros"]
	recurrent_initializer	categorical	string	["random_normal", "random_uniform", "truncated_normal", "glorot_normal", "glorot_uniform", "he_normal", "he_uniform", "identity", "orthogonal", "variance_scaling", "lecun_normal", "lecun_uniform", "zeros"]
	bias_initializer	categorical	string	["random_normal", "random_uniform", "truncated_normal", "glorot_normal", "glorot_uniform", "he_normal", "he_uniform", "variance_scaling", "lecun_normal", "lecun_uniform", "zeros"]
	unroll	catrgorical	boolean	[False]
	kernel_regularizer	categorical	string or null	[None, "l1", "l2"]
	bias_regularizer	categorical	string or null	[None, "11", "12"]
	activity_regularizer	categorical	string or null	[None, "11", "12"]
	recurrent_regularizer	categorical	string or null	[None, "l1", "l2"]
	use_bias	categorical	boolean	[True]
	mask_zero	categorical	boolean	[False, True]
	optimizer	categorical	string	["adadelta", "adagrad", "adam", "adamax", "ftrl", "nadam", "rmsprop", "sgd"]
	loss	categorizal	string	["sparse_categorical_crossentropy"]
	dropout_rate	continuous	float	[0.0, 0.7]
LSTM structure	n_layers <sup>a</sup>	continuous	int	[1, 10]
	first_layer_nodes <sup>b</sup>	continuous	int	[32, 128]
	last_layer_nodes <sup>c</sup>	continuous	int	[8, 64]

**Table A4.** Parameters and their values and ranges considered for EO hyperparameter search: the case of LSTM.

<sup>a</sup> Number of hidden layers. If set to 1, then last\_layer\_nodes parameter will be ignored. If this is greater than 2, then the number of nodes of the layers between the first and last will be computed using a linear function.

<sup>b</sup> Number of nodes for the first hidden layer. <sup>c</sup> Number of nodes for the last hidden layer.

# References

- Statista Research Department. Number of Global Social Network Users 2018–2022, with Forecasts from 2023 to 2027. 2022. Available online: https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/ (accessed on 14 September 2022).
- Zhao, J.; Liu, K.; Xu, L. Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Comput. Linguist. 2016, 42, 595–598. [CrossRef]
- 3. Gentzkow, M.; Kelly, B.; Taddy, M. Text as Data. J. Econ. Lit. 2019, 57, 535–574. [CrossRef]
- 4. Rust, R.T.; Rand, W.; Huang, M.H.; Stephen, A.T.; Brooks, G.; Chabuk, T. Real-Time Brand Reputation Tracking Using Social Media. *J. Mark.* 2021, *85*, 21–43. [CrossRef]
- 5. Praveen, S.; Ittamalla, R.; Deepak, G. Analyzing the attitude of Indian citizens towards COVID-19 vaccine—A text analytics study. *Diabetes Metab. Syndr. Clin. Res. Rev.* 2021, 15, 595–599. [CrossRef]
- 6. Bonifazi, G.; Breve, B.; Cirillo, S.; Corradini, E.; Virgili, L. Investigating the COVID-19 vaccine discussions on Twitter through a multilayer network-based approach. *Inf. Process. Manag.* **2022**, *59*, 103095. [CrossRef]
- Goldberg, Y. A Primer on Neural Network Models for Natural Language Processing. J. Artif. Intell. Res. 2016, 57, 345–420. [CrossRef]
- 8. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent Trends in Deep Learning Based Natural Language Processing [Review Article]. *IEEE Comput. Intell. Mag.* 2018, 13, 55–75. [CrossRef]
- Eisenstein, J. What to do about bad language on the internet. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, GA, USA, 9–14 June 2013; Vanderwende, L., Daume, H., III, Kirchhoff, K., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2013; pp. 359–369.
- Nguyen, D.Q.; Vu, T.; Nguyen, A.T. BERTweet: A pre-trained language model for English Tweets. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020—Demos, Online, 16–20 November 2020; Liu, Q., Schlangen, D., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 9–14. [CrossRef]

- 11. Cerruto, F.; Cirillo, S.; Desiato, D.; Gambardella, M.S.; Polese, G. Social network data analysis to highlight privacy threats in sharing data. *J. Big Data* 2022, *9*, 19. [CrossRef]
- Barrière, V.; Balahur, A. Improving Sentiment Analysis over non-English Tweets using Multilingual Transformers and Automatic Translation for Data-Augmentation. In Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain, 8–13 December 2020; Scott, D., Bel, N., Zong, C., Eds.; International Committee on Computational Linguistics: New York, NY, USA, 2020; pp. 266–271. [CrossRef]
- Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; Auli, M. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Ammar, W., Louis, A., Mostafazadeh, N., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 48–53. [CrossRef]
- 14. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv* 2019, arXiv:1910.03771.
- 15. Omar, N.O.; Harbaoui, A.; Ghezala, H.B. Opinion Mining and Sentiment Analysis on DEFT. Int. J. Cogn. Lang. Sci. 2021, 15, 54–57.
- Díaz-Galiano, M.C.; Vega, M.G.; Casasola, E.; Chiruzzo, L.; Cumbreras, M.Á.G.; Cámara, E.M.; Moctezuma, D.; Montejo-Ráez, A.; Cabezudo, M.A.S.; Tellez, E.S.; et al. Overview of TASS 2019: One More Further for the Global Spanish Sentiment Analysis Corpus. In Proceedings of the Iberian Languages Evaluation Forum Co-Located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, 24 September 2019; Cumbreras, M.Á.G., Gonzalo, J., Cámara, E.M., Martínez-Unanue, R., Rosso, P., Carrillo-de-Albornoz, J., Montalvo, S., Chiruzzo, L., Collovini, S., Gutiérrez, Y., et al., Eds.; CEUR-WS.org: Tilburg, The Netherlands, 2019; Volume 2421, pp. 550–560.
- Ciobotaru, A.; Dinu, L.P. RED: A Novel Dataset for Romanian Emotion Detection from Tweets. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), Online, 1–3 September 2021; Angelova, G., Kunilovskaya, M., Mitkov, R., Nikolova-Koleva, I., Eds.; INCOMA Ltd.: Moscow, Russia, 2021; pp. 291–300.
- Istrati, L.; Ciobotaru, A. Automatic Monitoring and Analysis of Brands Using Data Extracted from Twitter in Romanian. In Proceedings of the IntelliSys 2021: Intelligent Systems and Applications—Proceedings of the 2021 Intelligent Systems Conference, Amsterdam, The Netherlands, 2–3 September 2021; Lecture Notes in Networks and Systems; Arai, K., Ed.; Springer: Cham, Switzerland, 2021; Volume 296, pp. 55–75. [CrossRef]
- Tache, A.M.; Gaman, M.; Ionescu, R.T. Clustering Word Embeddings with Self-Organizing Maps. Application on LaRoSeDa—A Large Romanian Sentiment Data Set. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, 19–23 April 2021; Merlo, P., Tiedemann, J., Tsarfaty, R., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 949–956. [CrossRef]
- Cicada Technologies. Innovative Platform for Measuring TV Audience, Automatic Identification of Viewers and Correlating it with Analytic Data from Social Media. 2020. Available online: https://www.cicadatech.eu/projects/ (accessed on 9 September 2022).
- 21. technobium.com. Analiza de Sentiment Pentru Limba Romana. 2017. Available online: http://technobium.com/ (accessed on 26 August 2022).
- Lupea, M.; Briciu, A. Studying emotions in Romanian words using Formal Concept Analysis. *Comput. Speech Lang.* 2019, 57, 128–145. [CrossRef]
- Feraru, M.; Zbancioc, M. Emotion recognition using Lyapunov exponent of the Mel-frequency energy bands. In Proceedings of the 2014 6th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Bucharest, Romania, 23–25 October 2014; IEEE Computer Society: Washington, DC, USA, 2014; pp. 19–22. [CrossRef]
- 24. Feraru, S.M.; Schuller, D.; Schuller, B.W. Cross-language acoustic emotion recognition: An overview and some tendencies. In Proceedings of the 2015 International Conference on Affective Computing and Intelligent Interaction, ACII 2015, Xi'an, China, 21–24 September 2015; IEEE Computer Society: Washington, DC, USA, 2015; pp. 125–131. [CrossRef]
- Lupea, M.; Briciu, A.; Bostenaru, E. Emotion-based Hierarchical Clustering of Romanian Poetry. *Stud. Inform. Control* 2021, 30, 109–118. [CrossRef]
- Tufiş, D.; Barbu Mititelu, V. The Lexical Ontology for Romanian. In *Language Production, Cognition, and the Lexicon*; Gala, N., Rapp, R., Bel-Enguix, G., Eds.; Springer: Cham, Switzerland, 2015; pp. 491–504. [CrossRef]
- Balahur, A.; Turchi, M. Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis. *Comput. Speech Lang.* 2014, 28, 56–75. [CrossRef]
- Balahur, A.; Ortega, J.M.P. Sentiment analysis system adaptation for multilingual processing: The case of tweets. *Inf. Process. Manag.* 2015, 51, 547–556. [CrossRef]
- Balahur, A.; Turchi, M. Multilingual Sentiment Analysis using Machine Translation? In Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis, WASSA@ACL 2012, Jeju Island, Korea, 12 July 2012; Balahur, A., Montoyo, A., Martínez-Barco, P., Boldrini, E., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2012; pp. 52–60.
- 30. Medhat, W.; Hassan, A.; Korashy, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams Eng. J.* 2014, 5, 1093–1113. [CrossRef]
- 31. Kowsari, K.; Jafari Meimandi, K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D. Text classification algorithms: A survey. *Information* **2019**, *10*, 150. [CrossRef]

- 32. McCallum, A.; Nigam, K. A comparison of event models for naive bayes text classification. In Proceedings of the 1998 AAAI Workshop on Learning for Text Categorization, Madison, WI, USA, 26–27 July 1998; pp. 41–48.
- Joachims, T. Text categorization with support vector machines: Learning with many relevant features. In Proceedings of the 10th European Conference on Machine Learning—ECML-98, Chemnitz, Germany, 21–23 April 1998; Lecture Notes in Computer Science; Nédellec, C., Rouveirol, C., Eds.; Springer: Berlin/Heidelberg, Germany, 1998; Volume 1398, pp. 137–142. [CrossRef]
   Parimum L. Para Law Foundation Mark Learning 2001, 47, 5, 22 [CrossRef]
- 34. Breiman, L. Random Forests. Mach. Learn. 2001, 45, 5–32. [CrossRef]
- 35. Ng, A.Y.; Jordan, M.I. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In Proceedings of the Annul Conference on Neural Information Processing Systems [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001], Vancouver, BC, Canada, 3–8 December 2001; Dietterich, T.G., Becker, S., Ghahramani, Z., Eds.; MIT Press: Cambridge, MA, USA, 2001; pp. 841–848.
- Ojha, V.K.; Abraham, A.; Snásel, V. Metaheuristic design of feedforward neural networks: A review of two decades of research. Eng. Appl. Artif. Intell. 2017, 60, 97–116. [CrossRef]
- Jaderberg, M.; Simonyan, K.; Vedaldi, A.; Zisserman, A. Reading text in the wild with convolutional neural networks. *Int. J. Comput. Vis.* 2016, 116, 1–20. [CrossRef]
- Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16–21 June 2013; Volume 28, pp. 1310–1318.
- 39. Salton, G.; Buckley, C. Term-weighting approaches in automatic text retrieval. Inf. Process. Manag. 1988, 24, 513–523. [CrossRef]
- 40. Le, Q.V.; Mikolov, T. Distributed Representations of Sentences and Documents. In Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21–26 June 2014; Volume 32, pp. 1188–1196.
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of the 27th Annual Conference on Neural Information Processing Systems 2013: Advances in Neural Information Processing Systems 26, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.
- Jolliffe, I.; Cadima, J. Principal component analysis: A review and recent developments. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* 2016, 374, 20150202. [CrossRef] [PubMed]
- Pauca, V.P.; Shahnaz, F.; Berry, M.W.; Plemmons, R.J. Text Mining Using Non-Negative Matrix Factorizations. In Proceedings of the Fourth SIAM International Conference on Data Mining, Lake Buena Vista, FL, USA, 22–24 April 2004; Berry, M.W., Dayal, U., Kamath, C., Skillicorn, D.B., Eds.; SIAM: Philadelphia, PA, USA, 2004; pp. 452–456. [CrossRef]
- 44. Dumais, S.T. Latent semantic analysis. Annu. Rev. Inf. Sci. Technol. 2004, 38, 188–230. [CrossRef]
- 45. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Burstein, J., Doran, C., Solorio, T., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4171–4186. [CrossRef]
- Pires, T.; Schlinger, E.; Garrette, D. How Multilingual is Multilingual BERT? In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July–2 August 2019; Korhonen, A., Traum, D.R., Màrquez, L., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4996–5001. [CrossRef]
- Liu, W.; Zhou, P.; Wang, Z.; Zhao, Z.; Deng, H.; Ju, Q. FastBERT: A Self-distilling BERT with Adaptive Inference Time. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10 July 2020; Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 6035–6044. [CrossRef]
- Rosenthal, S.; Farra, N.; Nakov, P. SemEval-2017 Task 4: Sentiment Analysis in Twitter. In Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, BC, Canada, 3–4 August 2017; Bethard, S., Carpuat, M., Apidianaki, M., Mohammad, S.M., Cer, D.M., Jurgens, D., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 502–518. [CrossRef]
- 49. Barbieri, F.; Anke, L.E.; Camacho-Collados, J. XLM-T: A Multilingual Language Model Toolkit for Twitter. *arXiv* 2021, arXiv:2104.12250.
- Pota, M.; Ventura, M.; Fujita, H.; Esposito, M. Multilingual evaluation of pre-processing for BERT-based sentiment analysis of tweets. *Expert Syst. Appl.* 2021, 181, 115119. [CrossRef]
- Bokaee Nezhad, Z.; Deihimi, M.A. A Combined Deep Learning Model for Persian Sentiment Analysis. *IIUM Eng. J.* 2019, 20, 129–139. [CrossRef]
- 52. Bokaee Nezhad, Z.; Deihimi, M.A. Twitter sentiment analysis from Iran about COVID 19 vaccine. *Diabetes Metab. Syndr. Clin. Res. Rev.* 2022, 16, 102367. [CrossRef]
- 53. Dumitrescu, S.D.; Rebeja, P.; Lorincz, B.; Gaman, M.; Avram, A.; Ilie, M.; Pruteanu, A.; Stan, A.; Rosia, L.; Iacobescu, C.; et al. LiRo: Benchmark and leaderboard for Romanian language tasks. In Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, Virtual, 6–14 December 2021.
- 54. Avram, A.; Catrina, D.; Cercel, D.; Dascalu, M.; Rebedea, T.; Pais, V.F.; Tufis, D. Distilling the Knowledge of Romanian BERTs Using Multiple Teachers. *arXiv* 2021, arXiv:2112.12650.
- Feraru, S.M.; Teodorescu, H.; Zbancioc, M. SRoL—Web-based Resources for Languages and Language Technology e-Learning. Int. J. Comput. Control 2010, 5, 301–313. [CrossRef]

- Banea, C.; Mihalcea, R.; Wiebe, J. Multilingual Subjectivity: Are More Languages Better? In Proceedings of the COLING 2010: 23rd International Conference on Computational Linguistics, Beijing, China, 23–27 August 2010; Huang, C., Jurafsky, D., Eds.; Tsinghua University Press: Beijing, China, 2010; pp. 28–36.
- Honnibal, M.; Johnson, M. An Improved Non-monotonic Transition System for Dependency Parsing. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, 17–21 September 2015; Màrquez, L., Callison-Burch, C., Su, J., Pighin, D., Marton, Y., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; pp. 1373–1378. [CrossRef]
- 58. Aiken, M. An Updated Evaluation of Google Translate Accuracy. Stud. Linguist. Lit. 2019, 3, 253. [CrossRef]
- 59. Sequeira, L.N.; Moreschi, B.; Cozman, F.G.; Fontes, B. An Empirical Accuracy Law for Sequential Machine Translation: The Case of Google Translate. *arXiv* 2020, arXiv:2003.02817.
- 60. Kralj Novak, P.; Smailović, J.; Sluban, B.; Mozetič, I. Sentiment of Emojis. PLoS ONE 2015, 10, e0144296. [CrossRef]
- Řehůřek, R.; Sojka, P. Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, 22 May 2010; ELRA: Valletta, Malta, 2010; pp. 45–50. [CrossRef]
- 62. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830. [CrossRef]
- 63. Chollet, F. Keras. 2015. Available online: https://keras.io (accessed on 12 September 2022).
- Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for Hyper-Parameter Optimization. In Proceedings of the 25th Annual Conference on NIPS 2011: Advances in Neural Information Processing Systems 24, Granada, Spain, 12–15 December 2011; pp. 2546–2554.
- 65. Bergstra, J.; Bengio, Y. Random Search for Hyper-Parameter Optimization. J. Mach. Learn. Res. 2012, 13, 281–305. [CrossRef]
- Snoek, J.; Larochelle, H.; Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms. In Proceedings of the 26th Annual Conference on NIPS 2012: Advances in Neural Information Processing Systems 25, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 2960–2968.
- 67. Pelikan, M.; Goldberg, D.E.; Lobo, F.G. A Survey of Optimization by Building and Using Probabilistic Models. *Comput. Optim. Appl.* **2002**, *21*, 5–20. [CrossRef]
- Gorgolis, N.; Hatzilygeroudis, I.; Istenes, Z.; Gyenne, L. Hyperparameter Optimization of LSTM Network Models through Genetic Algorithm. In Proceedings of the 10th International Conference on Information, Intelligence, Systems and Applications, IISA 2019, Mumbai, India, 26–30 December 2019; Bourbakis, N.G., Tsihrintzis, G.A., Virvou, M., Eds.; IEEE: New York, NY, USA, 2019; pp. 1–4. [CrossRef]
- Violos, J.; Tsanakas, S.; Androutsopoulou, M.; Palaiokrassas, G.; Varvarigou, T. Next Position Prediction Using LSTM Neural Networks. In Proceedings of the SETN 2020: 11th Hellenic Conference on Artificial Intelligence, Athens, Greece, 2–4 September 2020; Spyropoulos, C.D., Varlamis, I., Androutsopoulos, I., Malakasiotis, P., Eds.; Association for Computing Machinery: Melbourne, Australia, 2020; pp. 232–240. [CrossRef]
- Bouras, I.; Aisopos, F.; Violos, J.; Kousiouris, G.; Psychas, A.; Varvarigou, T.A.; Xydas, G.; Charilas, D.; Stavroulas, Y. Mapping of Quality of Service Requirements to Resource Demands for IaaS. In Proceedings of the 9th International Conference on Cloud Computing and Services Science, CLOSER 2019, Crete, Greece, 2–4 May 2019; Muñoz, V.M., Ferguson, D., Helfert, M., Pahl, C., Eds.; SciTePress: Setubal, Portugal, 2019; pp. 263–270. [CrossRef]
- Mori, N.; Takeda, M.; Matsumoto, K. A comparison study between genetic algorithms and bayesian optimize algorithms by novel indices. In Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2005, Washington, DC, USA, 25–29 June 2005; Beyer, H., O'Reilly, U., Eds.; ACM: New York, NY, USA, 2005; pp. 1485–1492. [CrossRef]
- 72. Arenas Gomez, R. GASearchCV—Sklearn Genetic Opt Documentation. 2022. Available online: https://sklearn-genetic-opt. readthedocs.io/en/stable/api/gasearchcv.html (accessed on 14 September 2022).
- Fortin, F.; Rainville, F.D.; Gardner, M.; Parizeau, M.; Gagné, C. DEAP: Evolutionary algorithms made easy. J. Mach. Learn. Res. 2012, 13, 2171–2175. [CrossRef]
- Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; Krishnapuram, B., Shah, M., Smola, A.J., Aggarwal, C.C., Shen, D., Rastogi, R., Eds.; ACM: New York, NY, USA, 2016; pp. 1135–1144. [CrossRef]