STUDIA UNIVERSITATIS BABEŞ-BOLYAI OECONOMICA
VOLUME 70, ISSUE 1, 2025, pp. 83-111
DOI: 10.2478/subboec-2025-0005

BERTWEETRO: PRE-TRAINED LANGUAGE MODELS FOR ROMANIAN SOCIAL MEDIA CONTENT

Dan Claudiu NEAGU[*]
Babes-Bolyai University, Romania

Abstract: The introduction of Transformers, like BERT or RoBERTa, have revolutionized NLP due to their ability to better "understand" the meaning of texts. These models are created (pre-trained) in a self-supervised manner on large scale data to predict words in a sentence but can be adjusted (fine-tuned) for other specific NLP applications. Initially, these models were created using literary texts but very quickly the need to process social media content emerged. Social media texts have some problematic characteristics (they are short, informal, filled with typos, etc.) which means that a traditional BERT model will have problems when dealing with this type of input. For this reason, dedicated models need to be pre-trained on microblogging content and many such models have been developed in popular languages like English or Spanish. For under-represented languages, like Romanian, this is more difficult to achieve due to the lack of open-source resources. In this paper we present our efforts in pre-training from scratch 8 BERTweetRO models, based on RoBERTa architecture, with the help of a Romanian tweets corpus. To evaluate our models, we fine-tune them on 2 down-stream tasks, Sentiment Analysis (with 3 classes) and Topic Classification (with 26 classes), and compare them against Multilingual BERT plus a number of other popular classic and deep learning models. We include a commercial solution in this comparison and show that some BERTweetRO variants and almost all models trained on the translated data have a better accuracy than the commercial solution. Our best performing BERTweetRO variants place second after Multilingual BERT in most of our experiments, which is a good result considering that our Romanian corpus used for pre-training is relatively small, containing around 51,000 texts.

JEL classification: C45, C55, C88, O33

Keywords: machine learning; natural language processing; language models; transformers; text classification; under-resourced languages

[*] Corresponding author. Address: Faculty of Economics and Business Administration, Babeş-Bolyai University, 58-60, Teodor Mihali Street, 400591, Cluj-Napoca, Romania
E-mail: dan.neagu@econ.ubbcluj.ro

1. Introduction

A Transformer model is a deep learning architecture that uses a multi-head attention mechanism to transform texts into numerical representations called tokens. These are then converted into vectors using the word embedding table that is generated in the training stage and each vector is later contextualized with the help of the attention mechanism by using the scope of the context window paired with other (unmasked) tokens. Thus, the “intensity” of the important tokens is amplified but diminished for the less important ones (Vaswavi et al., 2017).

This methodology was proposed by researchers at Google in 2017 and is advantageous because it does not use recurrent units which means that the training times are lower than other architectures such as Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM). Transformers were developed initially to improve machine translation (Luong et al., 2015) but later they were used on a large scale in many sub fields of Natural Language Processing (NLP), Computer Vision (CV), and robotics.

In the area of NLP, Devlin et al. (2019) introduced Bidirectional Encoder Representations from Transformers (BERT) in 2018 which is based on transformers. This self-supervised learning model was revolutionary because it brought drastic improvements compared to the state-of-the-art models of that time and is now considered to be an early example of a Large Language Model (LLM). BERT is trained for masked token prediction and/or next sentence prediction on huge volumes of unlabeled texts using a deep bidirectional neural network design. The model learns the latent representations of tokens in their textual context bidirectionally allowing it to “understand” more nuanced and complex expressions. The original version of BERT is an “encoder-only” transformer architecture that has 4 modules: tokenizer (used to transform texts into a series of integers), embedding (used to convert sequences of tokens into a vector of real numbers), encoder (Transformer blocks with self-attention), and task head (used to decode the latent representation into token types). One big advantage of this type of model is that, once created, it can be fine-fined in a supervised manner on various downstream tasks such as question answering, document classification, and language translation with higher accuracy and efficiency.

A Robustly Optimized BERT Approach (RoBERTa) was created in 2019 by Liu et al. and is one of the most popular extensions of BERT. It preserves the original architecture but improves upon it by changing key hyperparameters, removing the next sentence prediction task, and by using larger mini-batch sizes in the pretraining process. These adjustments allows RoBERTa to handle complicated variations of language more easily thus, improving the performance that can be achieved in a wide number of applications.

The quality, quantity, and type of data used to create BERT or RoBERTa models from scratch affect their behavior in the field in which they are operated. For example, Raffel et al. (2020) has shown the importance of high-quality datasets by finding a direct correlation between the training sets and the results obtained by the models. The exponential growth of digitally created content and the reduction in computational costs have also made it easier to work with large scale datasets in order to improve state of the art performances in areas like emotion detection or part-of-speech (POS) tagging.

2. Literature review

Text data generated by users on social media platforms like Twitter, Facebook, or TikTok have some specific characteristics that are not usually found in literary texts or other standard documents such as articles, announcements, news, etc. These are referred to as “bad language” in the literature and include slang, jargon, non-standard abbreviations, grammatical mistakes, and an informal tone of expression (Eisenstein et al., 2023). Moreover, these texts are of a short nature due to the limitations imposed by social media platforms (for example Twitter has a limit of 280 characters for each tweet). Dealing with such issues is a mandatory and complex endeavor in any NLP task over social media content (Barriere et al., 2020).

The work of Dat Quoc Nguyen et al. (2020) titled “BERTweet: A pre-trained language model for English Tweets” is one of the biggest contributions in adapting a transformer to the unique characteristics of Twitter. BERTweet was created using public English tweets with the goal of providing better performances for downstream tasks like Sentiment Analysis (SA), Named Entity Recognition (NER), and Topic Modeling (TC). This paper showed how flexible this types of models are while also underscoring the importance of pretraining for unconventional textual contexts.

Other efforts were made to create custom BERT or RoBERTa models that could be used more successfully in niche domains and/or in underrepresented languages as a response to the limitations that come with generic pretrained models. Some of these limitations can be solved by fine tuning different architectures on data belonging to the target domain and by applying a pre-processing methodology that better suits the specialized tasks or linguistic features that are under study. For example, Beltagy et al. (2019) demonstrated that domain specific pretraining yields a better accuracy for custom named entity recognition and relation extraction in the field of biomedical text analysis. In the legal document understanding area, BERT-based models achieved decent results when tasked with parsing and analyzing contracts as shown in the work of Chalkidis et al. (2020). The more recent work of Conneau et al. (2019) introduced XLM-RoBERTa, a crosslingual language model trained on a multilingual corpus, and demonstrated its utility in a number of common NLP tasks in multiple languages.

With these advancements, researchers are now able to better capture the underlying linguistic patterns from texts but, as highlighted in Wei et al. (2021), augmenting the training data might still be required in some cases to address data scarcity and to increase resilience against noise/outliers.

3. Data

Both private institutions and researchers have a different number of options when it comes to the acquisition of data for training BERT models. The most accessible resources are of course the public text corpora which can be accessed and used by any actor to achieve practical or research goals in many domains. As a first example we can mention the Common Crawl[1] dataset which is a collection of web-scraped texts from the digital space and includes a variety of different languages and topics. Another important resource is the OpenAI WebText dataset[2] which is one of the biggest repository

[1] https://registry.opendata.aws/commoncrawl

[2] https://paperswithcode.com/dataset/webtext

of clean data often used for language understanding and generation (Radford et al., 2019). These datasets have been and still are extensively used to tweak language models, leading to a constant flow of improvements for the AI field.

If the available datasets don't meet some preset requirements then one may opt to collect their own dataset and contribute to the field by making it accesible to others. The TweetsCOV19[3] dataset is such an example, created during the COVID-19 pandemic it contains a vast number of Twitter posts that were used to analyze what kind of feelings and opinions people had about this difficult time period (Dimitrov et al., 2020). By doing so researchers are actively increasing the ammount and diversity of resources while also encouraging collaboration and reproducibility in the NLP community.

One may end up having access to proprietary or sensitive data case in which a number of aspects should be taken into account. These datasets are a rich source of information but the people who work with them need to follow and respect ethical guidelines and regulations in order to protect the users. There are certain techniques such as Federated Learning or Differential Privacy framework that can be used to mitigate the risk of privacy violation when working with sensible data (Erlingsson et al. 2019) and such additional efforts are expected to be made to ensure safety for all parties involved in the development life cycle of the models.

3.1. Twitter Stream

Twitter Stream[4], collected by Archive Team, is a valuable public corpus that offers a huge volume of texts that were scrapped from Twitter (now rebranded to "X") and stored in JSON format. This repo represents a testament of social media discourse and can be used for historic or other types of research as it covers all the years starting from 2012 until the middle of 2021, split into 2,900 files that amount to ≈ 6.8 TB of data.

The exact number of tweets in this dataset is not specified, but by considering the long-time frame that it covers plus the size of the documents we can say with a high degree of certainty that Twitter Stream should satisfy a large range of objectives. Researchers, private or public institutions could use this data to analyze trending topics, public sentiments, cultural or social events, and more in real time or in retrospect to answer questions about the dynamics of modern societies. Longitudinal studies can also be done to see how the writing style evolves over time because this archive contains the creation timestamp metadata of each post.

As opposed to other resources this one is not limited to include only tweets in internationally popular languages, such as English or Spanish, because in the web-scraping process the majority of public posts were collected, regardless of their language. Thus, we'll use the Twitter Stream to pretrain our custom BERT models as it captures the evolution of Romanian and the way it is used in a microblogging context.

However, Twitter texts are short and informal in nature, being filled with "bad language" elements such as slang, emojis, URLs, and hashtags which are common in the vast majority of online social platforms. These modern "flavours" in the way people communicate online are also found in Romanian texts and this is another reason for why

[3] https://data.gesis.org/tweetscov19/

[4] https://archive.org/details/twitterstream

we consider Twitter Stream to be suitable for the creation of one or more robust RoBERTa models that can address these issues in order to improve the performance of various down-stream NLP applications like sentiment analysis and topic classification.

3.2. Methodology

Given the size of the entire tweet archive, totaling close to 7 TB of raw data, our limited hardware dictates a need for data selection in order to train multiple versions of RoBERTa models in a reasonable timeframe. With respect to this, we decided to only use a subset of Twitter Stream that encompasses approximately 800 GB of data spanning over the course of one year: July 2020 through June 2021.

Another factor that made us take this action is related to the fine-tuning tasks that are going to be made on the newly created RoBERTa models, namely Sentiment Analysis and Topic Classification, and by acknowledging this constraint we want to state that our aim is to establish a Proof of Concept (POC) that can demonstrate the feasibility of training BERT based models on Romanian social media texts using a relatively small dataset. By doing this we'll most likely achieve lower performances when compared to using the whole archive as the training set, but our ultimate goal is to show that it's possible for researchers to create decent models in cases where there are strong hardware or time limitations. In future iterations, if additional computational resources become available to us, we would like to integrate the rest of the data in the pretraining pipeline to learn even more powerful models.

As a first step we downloaded the data belonging to the target period mentioned in the previous paragraphs after which we performed a manual inspection to familiarize with the structure and nature of it. The data is presented in JSON documents that contain two different types of instances, one denoting the removal of content from the platform and includes the ID of the deletion plus some other metadata but without any other textual information. The other type, referred to as "post", contains a lot of information but of interest to our study are the "text" field, which represents the tweet message, and the "lang" field, which indicates the language of the message.

Next, we selected and examined in more detail 200 random posts from a two-month period. During this we discovered a major problem with "lang": a number of tweets were labeled as Romanian when in reality they weren't. Many were simply misclassified, in some extreme cases as a very different language like Malay, and others were pure "noise" posts that only contained a mix of Twitter mentions, hashtags, URLs, and emojis which makes them unusable for our study. This highlights the problems that can appear when dealing with online user generated content where the informal tone of communication, errors in grammar, and other irregularities are degrading the accuracy of automated language identification tools.

Following this initial investigation we decided to use Python[5] together with *langid*[6] to correctly identify the language of the posts. We selected this library because it has been trained on a large number of languages (currently supporting 97 in total) which makes it a good choice for our multilingual dataset and additionally it offers very fast processing times paired with state-of-the-art results. Another advantage of lanid compared to other algorithms is that it offers a "confidence level" score for each prediction that acts as a measure of reliability.

[5] https://www.python.org/

[6] https://pypi.org /project/langid/

We ran langid on the same subset of 200 tweets using a high threshold approach in which we consider the texts to be Romanian only if the confidence level exceeds 95% to avoid the incorporation of false positives in our corpus. We made a second review of the language classification and saw that most of the texts were labeled correctly this time around but some outliers still persisted.

The overall performance on the raw texts can be considered satisfactory but to have even better results we decided to implement a preprocessing pipeline that includes the automatic identification and removal of URLs, Twitter mentions, Twitter hashtags, and emoticons from the tweets. With this mechanism in place we want to deliver cleaner and more standardized texts to langid in the hope of improving the accuracy.

We ran langid once again but this time on the cleaned data and performed another round of investigations. The results were clearly better which means that the proportion of tweets correctly labeled as Romanian has increased, thus validating our custom language identification framework. Due to this we'll make use of this preprocessing pipeline in the later stages of model development when needed.

Table 1 shows that over a period of 12 months we identified and extracted around 51,000 tweets posted in Romanian, which means that we have ≈ 4,250 tweets for each month on average. Because of preprocessing and language identification the total execution time for this extraction process has very high, totaling to over 72 hours.

Table 1: Comparison of Romanian tweets

Year-Month	Number of texts labeled as Romanian in Twitter Stream	Number of texts labeled as Romanian with our approach	Percent Romanian	Execution Time (Hours)
2020-07	48415	4256	8.8	6.4
2020-08	56292	5100	9.06	7.7
2020-09	59346	4729	7.97	7.3
2020-10	57778	4788	8.27	7.5
2020-11	48867	4406	9.02	5.5
2020-12	52896	4935	9.33	6.1
2021-01	22621	1771	7.83	2.5
2021-02	56163	4621	8.23	6.21
2021-03	57993	5210	8.98	6.7
2021-04	24149	2095	8.68	2.7
2021-05	58576	4702	8.03	7.2
2021-06	52475	4330	8.25	6.3

The size of this data might seem pretty modest, but it does align with the low number of Romanian Twitter users. With Statista[7] as source we found out that the number of Twitter users in Romania was around 600,000 during the time period targeted by us. It is also important to note that not all users make their posts public and additionally some accounts might have privacy settings in place. These aspects together with certain geographical or other restrictions mean that part of the generated content may have been overlooked/skipped during the scraping of Twitter Stream.

[7] https://www.statista.com/forecasts/1143811/twitter-users-in-romania

As a short summary, even though that our dataset has 51,000 tweets and could be viewed as small at a first glance we argue that it's sufficient to provide a relevant snapshot of the activity of Romanian speakers on Twitter. In the next sections we will use this dataset to train from scratch a number of BERT models but other researchers can employ it for any other type of purposes.

4. BERTweetRO

Researchers and private institutions have realized how important linguistic diversity is and the need to have solutions capable of addressing the challenges of less popular languages that face a scarcity of digital resources such as training sets or custom lexicons/dictionaries. Thus, in the last period of time an increasing number of efforts in pretraining transformer based models for underrepresented languages has been observed. There is also the option of creating multilingual models that cover several languages, an approach that brings very good results, but in some use cases it has been found that monolingual models fine-tuned on specific down stream tasks may offer superior performances (Velankar et al., 2022).

For Romanian several studies have tackled the task of creating language models by leveraging the transformer architecture together with large scale datasets to increase the level of automated language understanding and generation. Here we can mention the works of Dumitrescu et al. (2020) who introduced the first purely Romanian transformer-based language model which outperformed Multilingual BERT in the NER task and Masala et al. (2020) who created RoBERT using random texts crawled from the internet and formal texts from Romanian Wikipedia pages.

4.1 Variants

We want to develop 8 distinct RoBERTa variants in total and the motivation behind this is based on the linguistic diversity and complexity of Romanian as well as the varying preprocessing steps that might be needed in some NLP applications. The factors for investigation that we consider to be most important are: text case sensitivity, custom text preprocessing, and the number of tokens. We cover all these aspects in order to increase our chances of finding a model that is truly capable of handling social media texts in real life applications.

BERTweetRO model variants:

- Raw Cased
- Raw Uncased
- PreProcessed (PP) Cased
- PreProcessed (PP) Uncased
- Min Tokens Raw Cased
- Min Tokens Raw Uncased
- Min Tokens PreProcessed (PP) Cased
- Min Tokens PreProcessed (PP) Uncased

The first four variants from the list (Raw Cased, Raw Uncased, PreProcessed Cased, and PreProcessed Uncased) differ from one another in the preprocessing steps and text case handling. Raw Cased preserves the original casing, Raw Uncased converts all characters to lowercase while the PP Cased and PP Uncased variants transform the data by removing all the URLs, Twitter mentions and hashtags, emoticons, and platform reserved keywords; with the former keeping the original text case and the latter converting to all lowercase. These are the main contenders for our experiments that will allow us to see what impact (if any) case sensitivity and preprocessing has on the models created in this fashion.

The next four variants (Min Tokens Raw Cased, Min Tokens Raw Uncased, Min Tokens PreProcessed Cased, and Min Tokens PreProcessed Uncased) are similar to the first ones, the difference being that in these cases we exclude the tweets that have less than five tokens/words from the dataset. With this filtration we want to remove as many noisy instances as possible from the training set in the hope of improving the predictive power of the models.

In the end we'll compare these variants against each others to find the ones that deliver the best results.

4.2 Tokenizer training

Lexical tokenization is the process in which a text is transformed on a semantic or syntactic basis into a number of meaningful lexical tokens that belong to a predefined category. A common category employed for this is the part of speech which includes nouns, verbs, adjectives, punctuation marks, etc. The tokenization used in the case of transformers or large language models is similar to lexical tokenization but differs in two ways. First of all, lexical tokenization is based usually on a lexical grammar while LLM tokenizers use probability approaches. Secondly, LLM tokenizers include an additional procedure in which textual tokens are transformed to numbers (Alfred et al., 2007).

In other words, tokenization can be seen as the bridge that connects the natural representation of the texts used as inputs and the numerical values that encode the information such that it can be used by machine learning models. Besides breaking down text into individual components, like words or sub-words, tokenizers are also tasked with assigning an unique ID to each token in order to increase processing speeds.

The simplest word based tokenizers are the Bag-Of-Words (BoW) model (Zhang et al., 2010) which uses a representation of the text in the form of a list of unordered tokens meaning that it disregards word order, and thus most of the syntax, but captures multiplicity and Term Frequency–Inverse Document Frequency (TF-IDF) which is an improvement over BoW because this method can measure the importance of a word to a document from a collection of documents adjusted to the fact that some words appear more frequently over the entire corpus (Leskovec et al., 2020). More complex algorithms such as Word2Vec (Goldberg et al., 2014) or fastText (Athiwaratkun et al., 2018) can capture the meaning of words based on the context of other words in their proximity and for these reasons, they use a multidimensional encoding in which each token is represented by a distinct vector, but this adds complexity to model training and interpretation. Each tokenizer has its own advantages and disadvantages in terms of vocabulary size, sub-word granularity, execution speeds, which means that researchers must choose and/or adjust the right method depending on task requirements, text format, and language characteristics.

Another important tokenizer is the Byte Pair Encoding (BPE) algorithm which encodes string of texts into a tabular form and it's commonly used in various downstream modeling tasks (Gage et al., 1994). A modification to the original algorithm was made allowing it to combine tokens that encode both single characters, including single digits or punctuation marks, and full words (Brown et al., 2020). In this case, all unique characters are considered to be an initial set of 1-character long n-grams. Next, the most frequent adjacent pairs of characters are merged to create new 2-character long n-grams and all instances of previous pairs are replaced by this new token. This process is repeated until a vocabulary of a predetermined size is reached. This version of BPE is very often set as the encoding method of LLMs and transformers. In contrast, the standard BPE doesn't merge the most frequent pair of bytes of data but instead replaces them with a new byte that was not seen in the initial dataset (Paass et al., 2023).

Due to the popularity and effectiveness of BPE we decided to apply it in our work with the help of the ByteLevelBPETokenizer implementation from HuggingFace[8] library.

To train each variant of BERTweetRO Tokenizer we selected the following parameter configuration:

- Vocabulary size of 16,000 tokens
- Minimum frequency threshold of 2
- A set of special tokens containing <s>, <pad>, </s>, <unk>, and <mask>

The special tokens used in the training process have the following meaning: <s> marks the beginning of a sequence and is used when models require a clearly defined starting point for the input sequences, <pad> is a padding token used to ensure that all sequences have the same size without adding any meaningful content and it is necessary because the sequences can have variable sizes but the models expects them to have the same size, </s> marks the end of a sequence and is used when the models require clearly defined ending points for the input sequences, <unk> is used to represent words or subwords that are not in the tokenizer's vocabulary to handle unknown inputs, and <mask> is used in the Masked Language Modeling (MLM) pretraining process where a number of tokens from the sequence are replaced with this value in order to train the model to predict the original token (task also known as "fill in the blanks").

The creation of the tokenizers consists in training them to transform the corpus of Romanian tweets in a number of ways that matches our target data variants: Raw Cased, Raw Uncased, PP Cased, PP Uncased, Min Tokens Raw Cased, Min Tokens Raw Uncased, Min Tokens PP Cased, and Min Tokens PP Uncased. During this process the BPE algorithm discovers and learns statistical patterns based on the input texts and iteratively updates its vocabulary to capture as much information as possible for each subword unit. The resulting 8 tokenizers models were then saved future usage.

4.3 Model training

To successfully learn our RoBERTa models for Romanian text processing we selected an internal configuration that can yield good performances in relation to the training times and we integrated the previously trained tokenizers with the eight variants of RoBERTa in a consistent way to ensure that the hyperparameters and the end-to-end system allows for a fair comparison of performances in the downstream tasks. We decided to use the approach called Masked Language Modeling (MLM),

[8] https://huggingface.co/

implemented with the help of Hugging's RobertaForMaskedLM, which is a pre-training technique that enables transformers to predict masked tokens from input sequences. This is done without the need for labeled data making it an unsupervised learning method and unlike other traditional algorithms that can only predict the next token in a given sequence MLM can use both the previous and following tokens to predict a masked one. Thus, the models that use MLM can better understand the context that surrounds each word and it was found that more diverse training objectives are generally better for overall model behaviour (Tay et al., 2022).

The architectural specifications of our RoBERTa models are as follows:

- Hidden size of 768
- 12 attention heads
- 12 hidden layers
- MLM probability of 15%

This selection of parameters was made in such a way as to balance predictive performance with computational efficiency in the hope that the models can still capture complicated patterns while remaining manageable for running on our hardware. The reason not all tokens are masked is to avoid the dataset shift problem that arises when the distribution of tokens seen during training differs greatly from inference. The vocabulary size is different for each individual model, being set by the associated tokenizer to ensure compatibility.

All variants were trained over 5 epochs as we observed that it's sufficient to lead to an acceptable level of convergence without costing too much in terms of execution time. A larger number of epochs could incrementally improve the performance but we decided against this in order to avoid the risk of overfitting. The masked language model probability of 15% is in line with the recommendations in the literature (He et al., 2020; Levine et al., 2020; Izsak et al., 2021), based on the reasoning that models can't learn good representations when too much text is masked and the training is inefficient when too little is masked. If the training set is extremely big (not the case for our experiments) then higher percentage values should be considered (Wettig et al., 2022).

The models were trained on our GPU with a batch size of 16 and the total execution time for all 8 variants was a little under 4 hours which is decent if we consider the high computational overhead that is expected when creating transformer models from scratch.

5. Fine-tuning

Fine tuning is a transfer learning approach in which the parameters of a pretrained model are adjusted on a new dataset in order to refine or enrich its functionality (Zhang et al., 2021). This can be done on the entire neural network or only on a subset of layers in which case the layers that are not fine-tuned are "frozen" i.e. they are not changed in the backpropagation step. A model can also be augmented with the help of "adapters" that contain a much smaller number of parameters compared to those of the original model and thus it's fine tuned in a more efficient way because the initial weights remain the same (Liu et al., 2022).

For some architectures like CNNs it is common to keep the first layers (the ones closest to the input layer) frozen as they have the role of capturing low level features while the last layers are fine-tuned because these often discern high level features that are more related to the task that the model is trained on (Zeiler et al., 2014). Large scale models that have been pretrained on extensive corpora are usually fine-tuned by reusing the original parameters as a starting point, on top of which task specific layer(s) that are trained from scratch are added. The alternative of fine-tuning the whole system is also an option that usually delivers superior results but it's more computationally demanding due to the larger number of parameters that must be adjusted to the downstream task (Dingliwal et al., 2021).

In the case of BERT, a pretrained model is used as the feature extraction module with the aim of capturing the general linguistic representations while the newly added task specific layers are trained on labeled data to handle the target assignment. In the works of Devlin et al. (2019) and Liu et al. (2019) fine tuning general language models to become specialized in other tasks is highlighted as an easy way of bringing new capabilities to existing models, or improve their current performance, with minimal additional training data. The most common NLP applications targeted by fine tuning are text classification, named entity recognition, part of speech tagging, and machine translation.

5.1. Sentiment Analysis

Sentiment analysis, also known as opinion mining or emotion AI, is the use of natural language processing, text analysis, computational linguistics, and biometrics to automatically identify, extract, quantify, and study affective states and subjective information. It is widely employed in various domains such as reviews and survey responses, online and social media, and healthcare materials. The application of sentiment analysis ranges from marketing and customer service to clinical medicine. With the rise of transformer-based models and deep language models, more difficult data domains can be analyzed, such as texts where authors typically express their opinion/sentiment in a less explicit fashion (Hamborg et al., 2021).

A popular sentiment analysis task is to classify the overall polarity of a given text into representative categories like “negative”, “neutral”, or “positive”. Depending on the context and requirements this can be done at the document, sentence, or feature/aspect level. The advantage of this approach relies on its simplicity and clear categorization process which makes it easy to understand and apply in practice. Its disadvantage is that it cannot capture nuanced emotions and to overcome this limitation the “beyond polarity” sentiment classification can be used. In this case more subjective emotional states such as enjoyment, anger, disgust, sadness, fear, and surprise (Ho et al., 2020) are considered. In some fields, i.e. mental health analysis, this method can provide a deeper and better understanding of the emotions that are expressed in written texts. Due to its complexity and subjectivity this technique works best when advanced models are employed and when richer datasets, which are hard to get, are used for training.

To fine tune for this scenario new layers need to be added on top of pretrained BERT or RoBERTa model after which the entire architecture is trained in a supervised manner on a dataset annotated with sentiments. This allows the model to learn and identify sentiment related features from the data and to make predictions on never seen before texts based on the learned patterns. The quality of the model depends, among others, on the volume of data, the optimization process, and the number of iterations used during training.

When talking about underrepresented languages, such as Romanian, we can specify the work of Ciobotaru et al. (2023) in which the authors trained a fastText-based model and fine-tuned a standard BERT-based model then compared their performances.

They selected a public dataset which contains COVID-19 related Twitter posts split in 2 categories (negative and positive). Next, they built upon this dataset by adding the “neutral” sentiment class and by adding more text samples to all classes. This new dataset was used as the benchmark in their experiments and the reported results on the test set showed that BERT achieved a macro F1 score of 0.84 while fastText had a worse score of 0.73. An additional temporal study over a period of 21 weeks was made in which the authors observed that the general opinion about COVID-19 vaccination changed from positive to negative. The last half of this time period also generated more debate among users resulting in a serious increase in the number of posts made.

We searched in a number of online platforms including academic databases and NLP repositories (like Kaggle) but couldn’t find a dataset that could match our requirements. In this work we want to apply a multinomial sentiment analysis on Romanian social media content with negative, neutral, and positive as the polarity classes. We did find some review datasets that contain sentiments about products (Briciu et al., 2024; Istrati et al. 2021) but these are not suitable for us due to the obvious differences between review data and social media posts. Also it is worth mentioning that most of these works only offer a binary analysis of sentiment (negative vs positive).

Thus, we decided to employ an open-source English dataset, translate it to Romanian using an automated translation service and use it as a “surrogate” resource in our experiments (Neagu et al., 2022).

For this research, we selected the Twitter US Airline Sentiment Tweets dataset[9], collected in 2015 and each tweet was manually labeled by external contributors with its global sentiment polarity (positive, negative and neutral). This data contains approximately 15,000 tweets with a class distribution as follows: 63% negative, 21% neutral, and 16% positive. Each tweet is also accompanied by the contributor’s confidence about the annotated sentiment and each negative tweet includes a reason for the assessment.

Next we conducted a series of experiments using the newly translated Romanian dataset together with all of our 8 MLM RoBERTa variants. With Hugging Face’s BertClassifier we fine-tuned each model for sentiment analysis using the correct tokenizer as the encoding mechanism. We split the data into training and test sets with the training dataset consisting of approximately 11,000 instances while the testing dataset consisting of the remaining 3,700 instances, thus providing a standard 75-25% train-test split. This separation was made such that the class distribution between the train and test data remained similar. Moreover, the English and Romanian train and test data are identical in the sense that they contain the same set of instances. Depending on each model variant the associated preprocessing pipeline was executed in the same manner it was used in the pre-training stage in order to ensure that a fair comparison between the models can be made later on.

[9] https://www.kaggle.com/crowdflower/twitter-airline-sentiment

We will evaluate the performance of our 8 fine tuned variants of RoBERTa models in a comparative study in which we include a number of traditional classifiers from the area of classic machine learning and deep learning. The classic models are paired with TF-IDF encoding and comprise of Bernoulli Naive Bayes (NB), Support Vector Machine with a linear kernel (Linear SVM), Random Forest (RF), and Logistic Regression (LR). The selected deep neural network architectures are Deep Neural Network (DNN) with TF-IDF encoding, Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM) network both of them using Word2Vec as the encoding method.

We note here that all the traditional classifiers from our comparative study underwent a rigorous hyperparameter optimization process, with the help of evolutionary algorithms, in order to maximize their performance. Evolutionary optimization was selected because it can reach an adequate combination of values for the hyperparameters very quickly (Pelikan et al., 2002) and it has been show to outperform other approaches like Bayesian optimization (Mori et al., 2005). Another advantage of this technique is that the optimization can be done in all three types of search spaces (continuous, discrete, and categorical) regardless of the classifier on which the optimization is performed. Also there are many research works related to the metaheuristic design of neural networks (Ojha et al., 2017) in which the parameters or even the architectural structure (Bochinski et al., 2017) of deep learning models were identified with the help of genetic algorithms (Tani et al., 2021). Thus, it is important to highlight that the fine tuned models will be compared with these popular classifiers which have achieved peak performance on the selected dataset.

Additionally, we also compare the performance of our variants against Hugging Face's Multilingual BERT model which was fine-tuned on our translated dataset but without hyperparameter optimization due to the high execution times that are required. For this process we selected standard parameters and the associated Multilingual Tokenizer was used as the encoder as it can handle a vast number of languages, including Romanian. With this benchmark we want to offer valuable insights into the performance of our smaller RoBERTa models relative to a large scale pre trained multilingual language model that is widely used by researchers and private companies.

For our custom variants we added an extra sequential layer for classification that can handle the output of the pretrained layers and the expected sentiment class labels. As in the case of Multilingual BERT, our variants were not subjected to the hyperparameter optimization process due to time constraints. Instead, the parameters used for fine-tuning were chosen based on standard industry recommendations but also by considering the initial parameters that were used to create the models: batch size of 32, BERT hidden size of 768, classification hidden size of 75, a max token length of 80, Rectified Linear Unit (ReLU) as the activation function, and categorical cross-entropy as the loss function. The number of epochs, ranging from 2 to 10, that offers a decent level of accuracy was investigated and identified individually for each RoBERTa variant as well as for Multilingual BERT.

The results of our comparative analysis are presented in Table 2 in which the models are evaluated strictly on the test sets using 3 metrics often used in the literature (Macro F1, Weighted F1, and Accuracy) to allow us to assess the performance of the models from different points of view. Accuracy is the most reported score in research works because it shows what percentage of total predictions made are correct. The standard F-measure is more complex, computed as the harmonic mean between precision and recall. Weighted F1-score is a variation of this metric in which a weight is added to the predictions based on their distribution percentage with the goal of assigning a greater contribution for the classes that have more instances (Raschka et al., 2018). On the other hand, Macro F1 is computed as the arithmetic mean of class wise F-scores thus, it treats all classes as equally important no matter their frequency. In the case of imbalanced class labels Macro F1 can better measure performance because it's more punishing for the models that regularly misclassify under represented instances (Ganganwar et al., 2012).

Given the unbalanced nature of our datasets we set Macro F1 as the main measure of predictive performance. By using doing this we want to make sure that our evaluation treats each class in an equal manner in order to offer a correct interpretation of model effectiveness across all sentiments. Therefore we filtered Table 2 based on the Macro-F1 scores in descending order which means that the best results are at the top of the table.

Table 2: Sentiment analysis performance

Classifier	Encoding	Macro F1	Weighted F1	Accuracy
Multilingual BERT	Multilingual Tokenizer	74.81	80.50	80.99
BERTweetRO Raw Cased	BERTweetRO Tokenizer Raw Cased	72.11	78.40	78.74
BERTweetRO Raw Uncased	BERTweetRO Tokenizer Raw Uncased	72.07	78.33	78.66
Bernoulli NB	TFIDF	71.91	78.20	78.20
BERTweetRO Raw Min Tokens Cased	BERTweetRO Tokenizer Raw Min Tokens Cased	71.67	78.14	78.61
BERTweetRO Raw Min Tokens Uncased	BERTweetRO Tokenizer Raw Min Tokens Uncased	71.58	78.00	78.47
LSTM	Word2Vec	71.39	77.98	78.17
Linear SVM	TFIDF	70.54	77.47	78.36
DNN	Word2Vec	69.19	76.23	77.20
Logistic Regression	TFIDF	69.04	76.45	77.81
CNN	Word2Vec	68.67	76.00	77.69
BERTweetRO PP Min Tokens Cased	BERTweetRO Tokenizer PP Min Tokens Cased	64.21	73.10	72.86
BERTweetRO PP Cased	BERTweetRO Tokenizer PP Cased	43.84	59.40	64.50
BERTweetRO PP Uncased	BERTweetRO Tokenizer PP Uncased	42.35	58.73	64.01
Random Forest	TFIDF	38.17	54.71	65.20
BERTweetRO PP Min Tokens Uncased	BERTweetRO Tokenizer PP Min Tokens Uncased	25.62	47.98	62.42

We can see that Multilingual BERT outperformed all the other classifiers by having higher scores across all considered metrics but it's important to note that our best performing RoBERTa variants, namely BERTweetRO Raw Cased and BERTweetRO

Raw Uncased, achieved a similar performance. The differences between them and Multilingual BERT are fairly small with Macro F1 around 3% lower and the other 2 metrics around 2% lower. This outcome was somewhat expected if we take into account the difference in the scale of data used for pre-training as Multilingual BERT benefited from a much larger volume and diverse data whereas our variants were trained on a considerable smaller dataset (around 51,000 tweets).

Surprisingly, the BERTweetRO variants that were trained on texts with a minimum token constraint (Raw Min Tokens Cased and Uncased) also had competitive results that are only slightly below of those obtained by the 2 variants that used all the instances from the training set. This means that by limiting the data used to train the models, i.e. keeping only the texts with more than five tokens, the predictive performance that can be achieved is not reduced in a significant manner and additionally this can improve to some degree the execution speeds. Bernoulli NB, despite its simplicity, obtained good results placing it in between these 4 variants. With this exception the BERTWeetRO models that did not used the text preprocessing pipeline performed better than all the classic and deep learning models.

Another thing that we want to highlighted is the fact that regardless of the BERTweetRO variant, the ones that were trained on the data with the original text case (a.k.a. the Cased variants) have marginally better results than the equivalent variants that were trained on the data in which all characters have been converted to lower case (a.k.a. the Uncased variants). In the middle of the ranking we have LSTM, Linear SVM, DNN, Logistic Regression, and CNN with decent performances, making them suitable for usage in real scenarios.

At the bottom of the table, where the models with the worst results are placed, we have all the BERTweetRO variants that were paired with our custom text preprocessing module which clearly shows that this process negatively affected their predictive performances. This means that better BERT based models can be developed by simply pretraining and fine tuning them on raw social media data without the need for additional text cleaning or feature engineering and for this reason we want to warn other researchers about the risks of extensive preprocessing in contexts similar to ours. These models together with Random Forest had by far the lowest predictive performance meaning that they cannot be considered for sentiment analysis.

After this study we fine tuned BERTweetRO Raw Cased and BERTweetRO Raw Uncased on the whole US Airline Tweets dataset and saved them for future applications.

5.2 Topic Classification

Discovering abstract topics that occur in a collection of texts or documents could be done with either Topic Classification or Topic Modeling. Topic modeling is an unsupervised technique (Blei et al., 2012; Vayansky et al. 2020) that doesn’t require labeled data, while topic classification is a supervised one, where labeled data is needed for model training.

Topic modeling is a widely used statistical tool for extracting latent variables from large datasets, being well suited for textual data. Among the most used methods for topic modeling we can mention Probabilistic Latent Semantic Analysis (PSLA) and Latent Dirichlet Allocation (LDA) which state that a document is a mixture of topics, where a topic is considered to convey some semantic meaning by a set of correlated words, typically represented as a distribution of words over the vocabulary. Statistical techniques are then used to learn the topic components (topic-to-word distributions) and mixture coefficients (topic proportions) of each document. In essence PSLA, LDA, and other conventional topic models reveal topics within a text corpus by implicitly capturing the document-level word co-occurrence patterns (Boyd et al. 2008).

However, directly applying these models on short texts will suffer from the severe data sparsity problem, i.e. the sparse word co-occurrence patterns found in individual document (Hong et al., 2010). Some workarounds try to alleviate the sparsity problem with Albanese and Feuerstein (2021) aggregating a number of short texts to create a lengthy pseudo-document, its effectiveness being heavily data dependent. The Biterm Topic Model (Cheng et al., 2014) extracts unordered word pairs (i.e. biterms) occuring in short texts and the latent topic components are then modeled using these biterms. This method seems to perform better for short texts compared to other traditional approaches.

The main advantage of topic modeling methods is that they do not require labeled data, thus data collection becomes more accessible and could be done in a fully or partially automated manner. Despite its popularity, topic modeling is prone to serious issues with optimization, noise sensitivity, instability which can result in data which is unreliable (Agrawal et al., 2018), and some techniques are not representative of real-world data relationships (Blei et al., 2006). This is usually due to strong assumptions regarding key parameters in the calculation process and the inefficiency of many optimization methods (which often attempt to overcome uncertainty by performing many time-consuming iterations to determine the best parameters). For example, setting the optimum number of topics to be extracted is not trivial and human intervention is needed in order to set a relevant topic label to each identified topic. This is done based on the representative key words or phrases belonging to each topic.

If labeled training data is available then topic classification can overcome most issues related to the unsupervised nature of topic modeling by categorizing the texts into a number of predefined classes based on the subjects/themes in them. In this case, machine learning algorithms treat topic classification as a regular text classification problem: having a set of training records/instances $D = \{X_1, X_2, \dots, X_n\}$, where each record X_i represents a data point (i.e. document, paragraph, sentence, word) and is labeled with one of *k* distinct topic labels, the purpose is to build models that are capable of identifying text patterns based on the training records in order to predict the topics of never seen before texts with reliable accuracy rates. Unlike topic modeling, topic classification is easier to understand and evaluate which makes the assessment and comparison of models more straightforward.

Learning models on a small dataset with around 770 tweets distributed over 18 classes, Lee et al. (2011) achieved an accuracy of ≈65% with the multinomial Naive Bayes classifier and ≈62% with the standard SVM classifier. In another study, Rahman and Akter (2019) worked with 6,000 texts extracted from Amazon's product review corpus[10] distributed over only 6 very specific topics and achieved a very high classification rate of approximately 92% with NB, 82% with k-NN, and 79% with decision trees.

[10] https://jmcauley.ucsd.edu/data/amazon/

Zeng et al. (2018) proposed a hybrid approach that combines topic modeling with topic classification. They first extracted the most relevant latent features with topic modeling and then fed them into supervised algorithms like SVM, CNN, and LSTM. For the experiments they used the Twitter dataset released by TREC2011[11], which contains around 15,000 tweets, semi-automatically labeled into 50 topic classes. The highest accuracy of ≈9.5% was achieved by the CNN architecture and can be considered modest at best. Furthermore, they conclude that the topic modeling component did not improve the learning capabilities of the classifiers in any significant way.

Regarding topic classification for Romanian texts the existing research is more limited. Here we can mention the work of Vasile et al. (2014) who evaluated the capabilities of some classic machine learning models when applied to blog content. The data used in this study was extracted from 219 blogs, each instance being labeled with 1 topic class from a total of 9: "Activism", "Business and Finance", "Art", "Travel", "Gastronomy", "Literature", "Fashion", "Politics", and "Religion and Spirituality". The Sequential Minimal Optimization (SMO) and Complement Naive Bayes (CNB) performed the best, both reaching an accuracy of around 77.8%. A lower accuracy of 73.3% was achieved by k-Nearest Neighbors (k-NN) while the standard Naive Bayes (NB) had the worst accuracy of only 68.9%. Important to note that the authors used a very small dataset in their experiments which is problematic because it's unlikely that these results can be reproduced on larger evaluation sets.

We couldn't find any other relevant research works that target Romanian social media content. An explanation for this might be the unusual traits (Barriere et al., 2020; Eisenstein et al., 2013) of microblogging texts which pose a lot of problems for traditional NLP systems. Additionally, labeled datasets are also missing which means that we'll have to translate a suitable English dataset in order to create the training data needed for our topic classification experiments.

For this reason we selected the News Category Dataset[12] which contains 202,372 news headlines collected between 2012 up to 2018 from HuffPost[13], formerly The Huffington Post until 2017, an American news aggregator and blog with localized and international editions. The site offers news, satire, blogs, original content, and covers a variety of topics like politics, business, entertainment, technology, popular media, and more. Each record of the dataset contains the following attributes: *category* (41 categories), *headline*, *short_description*, *authors*, *date* (of the publication), and *link* (URL link of the article). There are a number of reasons why we selected this dataset as the benchmark for our experiments: (i) it contains short texts similar to those found on social media platforms, (ii) the topics are fairly general and the number of topics is large enough, (iii) the category of each article was manually labeled, (iv) high data volume, and (v) it was relatively recently collected.

For our classification problem we will focus only on the headline and short description attributes of the dataset, ignoring the authors and date of publication. Therefore, we merged the headline and the short description attributes and created a novel attribute named *text_merged*. The vast majority of merged texts contain between 94 and 254 characters, with the mean being ≈174 and the standard deviation almost

[11] http://trec.nist.gov/data/tweets

[12] https://www.kaggle.com/datasets/rmisra/news-category-dataset

[13] https://www.huffpost.com/

80 characters. This proves that the generated texts have the characteristics of short texts similar to those present in social media platforms (i.e. a Twitter tweet is limited to 280 characters).

We did an initial investigation on this data and encountered some problems with the distribution and granularity of the original 41 class labels. The top-3 most popular classes are: “POLITICS” which contains ≈16% of the records, “WELLNESS” which contains ≈9% of the records, and “ENTERTAINMENT” which contains ≈8% of the records. The least most popular 4 classes are: “COLLEGE”, “LATINO VOICES”, “CULTURE \& ARTS”, and “EDUCATION” each containing around 0.5% of the records, meaning that there is a significant class imbalance in the data.

Besides this imbalance, we also noticed that there are two inconsistencies related to the existing topics: a subset of them are overlapping while others are highly granular. For example, the categories “SCIENCE” and “TECH” are too specific and could be represented under a single label called “SCIENCE & TECH” while other classes have different labels but denote the same thing, for example “ARTS & CULTURE” and “CULTURE & ARTS”. To address these issues we decided to refine the labels of the dataset by clustering together overly granular and synonymous categories. At the end of this process the revised dataset contains 26 topics that are truly distinct and no class has less than 1% of record labels, meaning that the least popular class has more than 2,000 records. This should increase the performance of the models that will be trained later but at the same time it ensures consistency and coherence in our topic classification task. The new class feature was named *category_merged*. For additional details about this reconstructing process the readers are referred to (Neagu et al., 2023).

Fine tuning in this context involves the use of an existing pre-trained model and adapting it to classify the inputed texts based on the discussion topics they convey. This is done by adding task specific layers on top of the BERT or RoBERTa model after which the entire architecture is trained in a supervised manner on the annotated dataset. This allows the model to learn and identify topic related features from the data and to make predictions on never seen texts with the help of the learned representations. The ability of the model to recognize the abstract themes in text data depends, among others, on the volume of data, the optimization process, and the number of iterations during training.

Next, similar to Sentiment Analysis fine-tuning, we conducted a series of experiments using the newly translated Romanian dataset together with all 8 variants of our pretrained MLM RoBERTa models. With Hugging Face’s BertClassifier we fine tuned each model for topic classification using the correct tokenizer but before doing this we split the data into training and testing sets. The training set contains 75% of instances while the testing test contains the remaining 25%. This split was made such that the class distribution between training and testing remained similar. Moreover, the English and Romanian train and test data are identical in the sense that they contain the same set of instances. Depending on each model variant the associated preprocessing pipeline was applied in the same manner it was used in the pretraining stage in order to ensure that a fair comparison between the models can be made later on.

We'll evaluate the performance of our eight fine tuned variants of RoBERTa models in a comparative study in which we include a number of traditional classifiers from the area of classic machine learning and deep learning. The classic models are paired with TF-IDF vectors and comprise of Bernoulli Naive Bayes (NB), Support Vector Machine with a linear kernel (Linear SVM), and Random Forest (RF). The selected deep neural network architectures are Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) both of them using Word2Vec embeddings. As in the case of Sentiment Analysis, these models underwent a hyperparameter optimization process with the help of genetic algorithms. This means that our fine tuned variants will be compared against models that have peak performance on the selected dataset.

Additionally, we compare the performance of our variants against Hugging Face's Multilingual BERT model which was fine tuned on our translated dataset but without hyperparameter optimization due to the high execution times that are required. Instead we used standard parameters and the associated Multilingual Tokenizer as the encoding mechanism as it can handle a vast number of languages including Romanian. With this benchmark we want to offer valuable insights into the performance of our smaller RoBERTa variants relative to a large scale pretrained multilingual language model that is widely used today.

For our custom variants we added an extra sequential layer for classification on top of the existing architecture that can handle the output of the pre-trained layers and the expected topic labels. As in the case of Multilingual BERT our variants were not subjected to the hyperparameter optimization process due to time constraints. Instead the parameters used for fine tuning were selected based on industry recommendations but also by considering the initial parameters that were used to create the models from scratch: batch size of 32, BERT hidden size of 768, classification hidden size of 128, a max token length of 120, Rectified Linear Unit (ReLU) as the activation function, and categorical cross-entropy as the loss function. The number of epochs, ranging from 2 to 10, which leads to an acceptable level of accuracy was investigated and identified individually for each RoBERTa variant as well as for Multilingual BERT.

Unlike sentiment analysis, where the goal is to detect a text's global polarity, the difficulty of topic classification resides also in the big number of target classes which often overlap (Gentzkow et al., 2019; Liu et al., 2020). To overcome this issue, some authors (Gupta et al., 2014; Oh et al., 2017) use the Top-K accuracy instead of the standard one. Rather than classifying a text into a single class and comparing it to the a-priori label, the model will predict the *K* most probable classes and if the correct label is among them, we consider the text as being correctly classified. In our work we take this into account and report the standard accuracy (i.e. Top-1), as well as Top-2 and Top-3, evaluated strictly on the test set.

Table 3 shows the results of our comparative study which are filtered using the Top-1 accuracy in descending order meaning that the best models appear at the beginning. Here we can see Multilingual BERT in the first place with impressive Top-1, Top-2, and Top-3 accuracies of 72.63%, 85.56%, and 90.25% respectively. This result was expected if we consider the huge volume of data on which this transformer model was pretrained, allowing it to generate robust initial representations, even in the Romanian language, which are then easily adjusted for topic classification with the help of our translated dataset.

Table 3: Topic classification performance

Classifier	Encoding	Top-1 Acc.	Top-2 Acc.	Top-3 Acc.	Opt. (s)	Train (s)	Test (s)
Multilingual BERT	Multilingual Tokenizer	72.63	85.56	90.25	N/A	7498	157
Linear SVM	TFIDF	66.73	80.05	85.30	11803	45.91	0.042
BERTweetRO Raw Uncased	BERTweetRO Tokenizer Raw Uncased	66.14	79.21	84.93		8436	135
BERTweetRO Raw Min Tokens Uncased	BERTweetRO Tokenizer Raw Min Tokens Uncased	66.07	79.10	84.80		6899	157
BERTweetRO Raw Min Tokens Cased	BERTweetRO Tokenizer Raw Min Tokens Cased	65.78	79.02	84.79		6923	157
BERTweetRO Raw Cased	BERTweetRO Tokenizer Raw Cased	65.63	78.75	84.48		8442	132
Bernoulli NB	TFIDF	62.80	77.70	84.11	398	0.59	0.04
CNN	Word2Vec	61.66	74.05	79.28	36797	56.98	1.65
BERTweetRO PP Min Tokens Cased	BERTweetRO Tokenizer PP Min Tokens Cased	54.55	66.60	72.94		6046	137
LSTM	Word2Vec	53.50	65.59	72.39	63605	119.1	6.16
Random Forest	TFIDF	16.56	28.89	37	845	0.6	0.183
BERTweetRO PP Min Tokens Uncased	BERTweetRO Tokenizer PP Min Tokens Uncased	16.56	28.89	37		6019	135
BERTweetRO PP Cased	BERTweetRO Tokenizer PP Cased	16.56	28.89	37		6027	135
BERTweetRO PP Uncased	BERTweetRO Tokenizer PP Uncased	16.56	28.89	37		6022	135

In the race for the runner up position we have several models with similar scores across all three evaluation metrics, namely Linear SVM and the four Raw BERTweetRO variants. These classifiers reached Top-1 accuracies between ≈65% and ≈66%, which are good enough for real life applications. Note that as previously mentioned Linear SVM benefited from a complex process of hyperparameter optimization, conducted on a high performance computer, while the BERTweetRO variants were fine tuned with default parameters on standard hardware. As in the case of Sentiment Analysis, the BERTweetRO variants that didn't use the custom text preprocessing pipeline obtained better results than the other four variants that did. The difference between our best variants and Multilingual BERT is around 6% but can be considered decent given the relative small size of the data we used for pretraining (around 51,000 texts) with respect to the much larger corpora that was used to pretrain Multilingual BERT. This means that by employing a richer dataset and by applying hyperparameter optimization we could potentially enhance the performance of the BERTweetRO variants in future iterations.

BERTweetRO Raw Min Tokens Uncased and Cased also have competitive performances, as they had in Sentiment Analysis, which reconfirms that by restricting the data that is used for fine tuning, i.e. only including the texts with more than five tokens, the predictive performance that can be achieved is not downgraded and at the same time this can improve (to some degree) the execution speeds when training and executing the models.

Bernoulli NB and CNN are next, with slightly lower scores when compared to the previous models, both having a similar performance when talking about Top-1 accuracy but in the case of Top-2 and Top-3 CNN lags behind by a pretty noticeable margin. These classifiers should behave more or less the same when predicting the main topic of a text but Bernoulli NB is to be preferred if one considers the second and third most probable topics as being important to their use case.

BERTweetRO PP Min Tokens Cased and LSTM share fourth place in our ranking with modest results across all considered metrics. At the bottom of the table we have Random Forest and the four BERTweetRO variants that incorporated the text preprocessing module, all of them having by far the worst predictive performance, with a Top-1 accuracy of only 16.5%. This once again underscores the effectiveness of simply pretraining and fine tuning BERT models on raw social media text data without the need for any text cleaning or feature engineering. For this reason we want to warn other researchers about the risks of extensive text preprocessing for similar NLP tasks in which microblogging data is involved, as this may lead to worse results. Hence, this group of models are not viable for topic classification in production environments.

After these experiments the BERTweetRO Raw Cased and BERTweetRO Raw Uncased variants were fine tuned on the entire News Category dataset to increase their generalization power and saved for future use.

6. Assessing and comparing Sentiment Analysis performance on real cases

Given that the final purpose of our work 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 will evaluate on these test sets the best performing models as reported in previously in this work: Multilingual BERT, BERTweetRO Raw Uncased, Bernoulli NB, LSTM, and DNN. Additionally, we'll compare these models against a public third-party sentiment analysis tool for Romanian called Sentimetric[14] to see where we stand in relation to a commercially available solution.

The tweets were manually labeled by five human volunteers who were trained in advance on how this process should be carried out. Each volunteer expressed an opinion about the polarity of the tweet and the final sentiment was set as the one

[14] http://sentimetric.ro/

selected by the majority. Labeling statistics regarding how they assessed the polarity is presented in Table 4. We shall note that the labeling task seemed to be a difficult one even 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 of the 5 contributors reached a unanimous decision. Furthermore, the class distribution of these tweets is significantly different from that of the Twitter US Airline Sentiment Tweets (presented in the last row of the table).

Table 4: Manual labelling statistics of tweet polarity, including the number of tweets and corresponding percentages

Dataset	Negative	Neutral	Positive	Unanimous Annotation
Airline industry tweets	51 (46.5%)	36 (30%)	33 (27.5%)	43 (35.8%)
General tweets	45 (37.5%)	32 (26.6%)	43 (35.8%)	47 (39.2%)
Twitter US Airline Sentiment Tweets	63%	21%	16%	

As in the case of the fine tuning experiments we report Macro F1, Weighted F1, and Accuracy as the evaluation metrics for each classifier but seeing the imbalanced distribution of labels of both dateset we again have to set Macro F1 as the main measure of model performance.

Table 5 contains the predictive performance of our target models on the 120 real life Romanian tweets that relate to the aviation industry. In this case we can see that Bernoulli Naive Bayes (NB) achieved the highest Macro F1 score of 61.18% and in second place, with a marginally lower score, we have Multilingual BERT. This result is a little surprising considering that Multilingual BERT had better results on the evaluation set used in the fine tuning section but the success of NB could be attributed to the hyperparameter optimization process that it went through.

Table 5: Model performance on Romanian aviation industry-specific tweets

Classifier	Encoding	Macro F1	Weighted F1	Accuracy
Bernoulli NB	TFIDF	61.18%	63.11%	65%
Multilingual BERT	Multilingual Tokenizer	60.45%	63.38%	65.83%
BERTweetRO Raw Uncased	BERTweetRO Tokenizer Raw Uncased	54.57%	56.68%	60%
LSTM	Word2Vec	52.71%	55.18%	58.33%
DNN	Word2Vec	52.22%	54.9%	59.17%
Sentimetric		45.72%	46.99%	47.5%

The test data includes a small number of samples but despite this our BERTweetRO Raw Uncased variant managed to secure an honorable third place across all evaluation metrics. Even thought that it failed to surpass Bernoulli NB and Multingual BERT its performance is better than the deep learning LSTM and DNN models. The Macro F1 of 54.5%, which is around 6% lower than the best score, can be considered acceptable given that the humans volunteers also had difficulties when labeling the texts.

The most important thing that we want to highlight here is that all of our models outperformed Sentimetric. This shows the positive impact of using a custom methodology for training and validating ML models when compared to off the shelf solutions. It also confirms the value of domain specific knowledge for obtaining better results in such contexts as we fine tuned our models on tweets from the same domain.

In Table 6 we present the performance of the classifiers on the Romanian general tweets dataset, i.e. tweets that don't belong to a single specific industry. In this case things are a little different as Multilingual BERT achieved the best result with a Macro F1 of 55.22% followed closely by BERTweetRO Raw Uncased with a negligible difference in score of only 1%. In both this assessment and the previous one, the transformer models placed at the top which means that they're more reliable for sentiment analysis in practice.

Table 6: Model performance on Romanian general tweets

Classifier	Encoding	Macro F1	Weighted F1	Accuracy
Multilingual BERT	Multilingual Tokenizer	52.22%	54.17%	55.85%
BERTweetRO Raw Uncased	BERTweetRO Tokenizer Raw Uncased	51.35%	52.39%	54.17%
Bernoulli NB	TFIDF	48.48%	49.42%	48.33%
DNN	Word2Vec	48.16%	49.29%	50.83%
Sentimetric		46.16%	47.3%	49.17%
LSTM	Word2Vec	43.17%	44.29%	45.83%

On the other hand, Bernoulli NB and the DNN architecture have more modest results that place them in the middle of the ranking but more surprising is that LSTM delivered a significantly worse performance in this case, being behind all the other models, including the solution offered by Sentimetric. The reasons to why this happened requires future investigations but it's possible that the complexity of the neural network together with its sensitivity to the shape of input data could have affected its ability to correctly recognize the sentiment patterns from these samples.

An interesting detail that we want to point out is that the overall performance of the models on these generic tweets is lower compared to the aviation industry. This decline, which is more obvious for the classic and deep learning models, is a direct result of domain differences between the texts used for training and the ones used for evaluation. For Multilingual BERT and BERTweetRO the decline is less serious due to the fact that they were pretrained on varied data and thus managed to better adapt in this scenario.

For both domains our models' results are lower than those obtained on the translated test set because now the tweets are real ones, not translated, and their inherent characteristics differ, i.e. from a statistical point of view the sets are extracted from different statistical populations.

Nonetheless, as in the case of the first evaluation set, we note that all our models (with the exception of LSTM) have significantly outperformed the commercial solution that was selected as the benchmark for comparison. This once again validates the importance of custom fine tuning and model optimization in achieving superior results for Romanian sentiment prediction.

7. Discussion and further work

As a first idea that we consider for future work directions is the hyperparameter optimization for our BERTWeetRO models for both Sentiment Analysis and Topic Classification. Although our initial experiments that used the default parameters generated promising results, we could considerably increase the predictive performances on these downstream tasks by applying a well thought optimization process. The parameters that we'd like to explore and adjust are: learning rate, dropout rate, batch size, number of neurons per layer, and number of layers.

Other aspects that might be worth to investigate are different text preprocessing steps and the usage of data augmentation algorithms to better clean and enrich our labelled datasets. By doing this the classifiers would have access to additional examples in the training phase which in turn should increase their capability of understanding the meaning of texts based on the context of the words within them. We could also employ different types of encodings and tokenizers to see how these impact the performance and execution times of the models.

In this paper we presented the fine-tuning methodology on two popular NLP tasks but this can be relatively easily extended to other commonly requested tasks in the industry. A good candidate would be Named Entity Recognition (NER) for Romanian texts, which consists in identifying a set of entities and classifying them into categories such as names of people, names of organizations, locations, calendar dates, etc. By fine-tuning our models for NER we can offer another NLP functionality that has direct or indirect application in various domains/systems like entity linking, information extraction, and semantic search.

And last but not least, we would like to increase the size of our pretraining repository by adding more Romanian tweets that cover a wider range of linguistic patterns, expressions, and discussion topics to allow our transformer models to reason with a higher level of generalization which would directly improve their behaviour on any downstream task. The classifiers' accuracy may also be improved by incorporating new annotated datasets for SA and TC in the training or fine tuning stage and this can be done by simply translating English datasets as demonstrated in our study.

8. Conclusions

In conclusion, our work offers a number of contributions for the NLP of social media content in Romanian, a language which is highly under resourced in this area. First of all, we identified and curated an open source repository that contains public tweets posted from July 2020 up to, and including, June 2021. These tweets can be used by anyone who was an interest for studying the characteristics of Romanian conversations in a microblogging space. Using this data we pretrained from scratch 8 different variants of BERTweetRO models, all based on the RoBERTa MLM architecture, and their corresponding text tokenizers that can be used on either raw and preprocessed texts.

Next, we selected the following NLP downstream tasks to fine tune our models on: (i) Sentiment Analysis, which refers to the classification of texts into 3 polarity classes (negative, neutral, and positive) based on the feelings they express, and (ii) Topic Classification, which refers to the classification of texts into 26 distinct and generic discussion topics. We couldn't find any annotated datasets suitable to our specific research needs so we decided to apply an automated translation service on two English datasets to create the equivalent resources in Romanian. We then fine-tuned our BERTweetRO variants and the popular Multilingual BERT on the translated datasets and aimed to achieve the highest predictive performance possible by finding the optimal number of epochs for each model.

We implemented a comprehensive test bed in which the transformer models were compared against a number of well-known classic and deep learning ML algorithms that were trained for the same tasks using the same datasets. The results of these experiments show that Multilingual BERT is indeed the best option but some of our BERTWeetRO models achieved comparable performances thus highlighting their potential for improvement in the future with the help of hyperparameter optimization or data augmentation. Bernoulli NB, Linear SVM, and CNN also had good overall results, which means that they can be employed in practice, especially when computational resources are limited.

An important finding we want to emphasize is that the text preprocessing steps had a serious negative impact on the performance of the BERTweetRO variants that used them. Thus, we want to warn others about the risks of extensive preprocessing for similar applications where social media texts are involved.

In the end we collected and manually labeled with sentiment polarity two sample sets of real life Twitter posts, one containing tweets specific to the aviation industry and the other containing generic tweets. Then we executed on these datasets our best performing BERTweetRO variant together with Multilingual BERT and evaluated their predictive performances against a commercial classifier called Sentimetric. In both domains our models delivered better results therefore validating our custom methodology for developing language models.

Our pre-trained BERTweetRO models, together with the variants fine-tuned for Sentiment Analysis and Topic Classification, are open source and can be accessed online[15].

References

Agrawal, A., Fu, W., & Menzies, T. (2018). What is wrong with topic modelling? And how to fix it using search-based software engineering. *Information and Software Technology*, 98, 74-88.

Albanese, F., & Feuerstein, E. (2021). Improved topic modelling in twitter through community pooling. In *String Processing and Information Retrieval: 28th International Symposium*, SPIRE 2021, Lille, France, October 4–6, 2021, Proceedings 28 (pp. 209-216). Springer International Publishing.

Alfred, V. A., Monica, S. L., & Jeffrey, D. U. (2007). Compilers principles, techniques & tools. *pearson Education*.

Athiwaratkun, B., Wilson, A. G., & Anandkumar, A. (2018). Probabilistic fasttext for multi-sense word embeddings. *arXiv preprint arXiv*:1806.02901.

Barriere, V., & Balahur, A. (2020). Improving sentiment analysis over non-English tweets using multilingual transformers and automatic translation for data-augmentation. *arXiv preprint arXiv*:2010.03486.

[15] https://huggingface.co/dan-neagu

Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A pretrained language model for scientific text. *arXiv preprint arXiv*:1903.10676.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4), 77-84.

Blei, D., & Lafferty, J. (2006). Correlated topic models. *Advances in neural information processing systems*, 18, 147.

Bochinski, E., Senst, T., & Sikora, T. (2017). Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms. In *2017 IEEE international conference on image processing (ICIP)* (pp. 3924-3928). IEEE.

Boyd-Graber, J., & Blei, D. (2008). Syntactic topic models. *Advances in neural information processing systems*, 21.

Briciu, A., Călin, A. D., Miholca, D. L., Moroz-Dubenco, C., Petrașcu, V., & Dascălu, G. (2024). Machine-Learning-Based Approaches for Multi-Level Sentiment Analysis of Romanian Reviews. *Mathematics*, 12(3), 456.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877-1901.

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., & Androutsopoulos, I. (2020). LEGAL-BERT: The muppets straight out of law school. *arXiv preprint arXiv*:2010.02559.

Cheng, X., Yan, X., Lan, Y., & Guo, J. (2014). Btm: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering*, 26(12), 2928-2941.

Ciobotaru, A., & Dinu, L. P. (2023). SART & COVIDSentiRo: Datasets for Sentiment Analysis Applied to Analyzing COVID-19 Vaccination Perception in Romanian Tweets. *Procedia Computer Science*, 225, 1331-1339.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... & Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv*:1911.02116.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). 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*, volume 1 (long and short papers) (pp. 4171-4186).

Dimitrov, D., Baran, E., Fafalios, P., Yu, R., Zhu, X., Zloch, M., & Dietze, S. (2020). Tweetscov19-a knowledge base of semantically annotated tweets about the covid-19 pandemic. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 2991-2998).

Dingliwal, S., Shenoy, A., Bodapati, S., Gandhe, A., Gadde, R. T., & Kirchhoff, K. (2021). Prompt Tuning GPT-2 language model for parameter-efficient domain adaptation of ASR systems. *arXiv preprint arXiv*:2112.08718.

Dumitrescu, S. D., Avram, A. M., & Pyysalo, S. (2020). The birth of Romanian BERT. *arXiv preprint arXiv*:2009.08712.

Eisenstein, J. (2013). 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* (pp. 359-369).

Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., & Thakurta, A. (2019). Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 2468-2479). Society for Industrial and Applied Mathematics.

Gage, P. (1994). A new algorithm for data compression. *The C Users Journal*, 12(2), 23-38.

Ganganwar, V. (2012). An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, 2(4), 42-47.

Gentzkow, M., Kelly, B., & Taddy, M. (2019). Text as data. *Journal of Economic Literature*, 57(3), 535-574.

Goldberg, Y., & Levy, O. (2014). word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv*:1402.3722.

Gupta, M. R., Bengio, S., & Weston, J. (2014). Training highly multiclass classifiers. *The Journal of Machine Learning Research*, 15(1), 1461-1492.

Hamborg, F., Donnay, K., & Merlo, P. (2021). NewsMTSC: a dataset for (multi-) target-dependent sentiment classification in political news articles. *Association for Computational Linguistics (ACL)*.

He, P., Liu, X., Gao, J., & Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv*:2006.03654.

Ho, V. A., Nguyen, D. H. C., Nguyen, D. H., Pham, L. T. V., Nguyen, D. V., Nguyen, K. V., & Nguyen, N. L. T. (2020). Emotion recognition for vietnamese social media text. In *Computational Linguistics: 16th International Conference of the Pacific Association for Computational Linguistics*, PACLING 2019, Hanoi, Vietnam, October 11–13, 2019, Revised Selected Papers 16 (pp. 319-333). Springer Singapore.

Hong, L., & Davison, B. D. (2010). Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics* (pp. 80-88).

Istrati, L., & Ciobotaru, A. (2022). Automatic monitoring and analysis of brands using data extracted from twitter in Romanian. In Intelligent Systems and Applications: *Proceedings of the 2021 Intelligent Systems Conference* (IntelliSys) Volume 3 (pp. 55-75). Springer International Publishing.

Izsak, P., Berchansky, M., & Levy, O. (2021). How to train BERT with an academic budget. *arXiv preprint arXiv*:2104.07705.

Lee, K., Palsetia, D., Narayanan, R., Patwary, M. M. A., Agrawal, A., & Choudhary, A. (2011). Twitter trending topic classification. In *2011 IEEE 11th international conference on data mining workshops* (pp. 251-258). IEEE.

Leskovec, J., Rajaraman, A., & Ullman, J. D. (2020). Mining of massive data sets. *Cambridge university press*.

Levine, Y., Lenz, B., Lieber, O., Abend, O., Leyton-Brown, K., Tennenholtz, M., & Shoham, Y. (2020). Pmi-masking: Principled masking of correlated spans. *arXiv preprint arXiv*:2010.01825.

Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., & Raffel, C. A. (2022). Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35, 1950-1965.

Liu, X., He, P., Chen, W., & Gao, J. (2019). Improving multi-task deep neural networks via knowledge distillation for natural language understanding. *arXiv preprint arXiv*:1904.09482.

Masala, M., Ruseti, S., & Dascalu, M. (2020). Robert–a romanian bert model. In *Proceedings of the 28th International Conference on Computational Linguistics* (pp. 6626-6637).

Mori, N., Takeda, M., & Matsumoto, K. (2005). A comparison study between genetic algorithms and bayesian optimize algorithms by novel indices. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation* (pp. 1485-1492).

Neagu, D. C., Rus, A. B., Grec, M., Boroianu, M. A., Bogdan, N., & Gal, A. (2022). Towards sentiment analysis for romanian twitter content. *Algorithms*, 15(10), 357.

Neagu, D. C., Rus, A. B., Grec, M., Boroianu, M., & Silaghi, G. C. (2022). Topic Classification for Short Texts. In *International Conference on Information Systems Development* (pp. 207-222). Cham: Springer International Publishing.

Nguyen, D. Q., Vu, T., & Nguyen, A. T. (2020). BERTweet: A pre-trained language model for English Tweets. *arXiv preprint arXiv*:2005.10200.

Oh, S. (2017). Top-k hierarchical classification. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 31, No. 1).

Ojha, V. K., Abraham, A., & Snášel, V. (2017). Metaheuristic design of feedforward neural networks: A review of two decades of research. *Engineering Applications of Artificial Intelligence*, 60, 97-116.

Paaß, G., & Giesselbach, S. (2023). Pre-trained Language Models. In *Foundation Models for Natural Language Processing: Pre-trained Language Models Integrating Media* (pp. 19-78). Cham: Springer International Publishing.

Pelikan, M., Goldberg, D. E., & Lobo, F. G. (2002). A survey of optimization by building and using probabilistic models. *Computational optimization and applications*, 21, 5-20.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8), 9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140), 1-67.

Rahman, M. A., & Akter, Y. A. (2019). Topic classification from text using decision tree, K-NN and multinomial naïve bayes. In *2019 1st international conference on advances in science, engineering and robotics technology* (ICASERT) (pp. 1-4). IEEE.

Raschka, S. (2021). Model evaluation, model selection, and algorithm selection in machine learning. *arXiv 2018. arXiv preprint arXiv*:1811.12808.

Tani, L., Rand, D., Veelken, C., & Kadastik, M. (2021). Evolutionary algorithms for hyperparameter optimization in machine learning for application in high energy physics. The European Physical Journal C, 81, 1-9.

Tay, Y., Dehghani, M., Tran, V. Q., Garcia, X., Wei, J., Wang, X., ... & Metzler, D. (2022). Ul2: Unifying language learning paradigms. *arXiv preprint arXiv*:2205.05131.

Vasile, A., Rădulescu, R., & Păvăloiu, I. B. (2014). Topic classification in Romanian blogosphere. In *12th Symposium on Neural Network Applications in Electrical Engineering* (NEUREL) (pp. 131-134). IEEE.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Vayansky, I., & Kumar, S. A. (2020). A review of topic modeling methods. *Information Systems*, 94, 101582.

Velankar, A., Patil, H., & Joshi, R. (2022). Mono vs multilingual bert for hate speech detection and text classification: A case study in marathi. In *IAPR workshop on artificial neural networks in pattern recognition* (pp. 121-128). Cham: Springer International Publishing.

Wei, J., Garrette, D., Linzen, T., & Pavlick, E. (2021). Frequency effects on syntactic rule learning in transformers. *arXiv preprint arXiv*:2109.07020.

Wettig, A., Gao, T., Zhong, Z., & Chen, D. (2022). Should you mask 15% in masked language modeling?. *arXiv preprint arXiv*:2202.08005.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference*, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13 (pp. 818-833). Springer International Publishing.

Zeng, J., Li, J., Song, Y., Gao, C., Lyu, M. R., & King, I. (2018). Topic memory networks for short text classification. *arXiv preprint arXiv*:1809.03664.

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2023). Dive into deep learning. *Cambridge University Press*.

Zhang, Y., Jin, R., & Zhou, Z. H. (2010). Understanding bag-of-words model: a statistical framework. *International journal of machine learning and cybernetics*, 1, 43-52.

Zhao, J., Liu, K., & Xu, L. (2016). Sentiment analysis: Mining opinions, sentiments, and emotions.