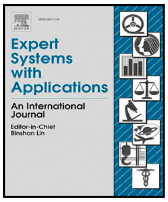
Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Expert Systems with Applications
An International Journal
Editor-in-Chief
Binshan Lin

TurkishBERTweet: Fast and reliable large language model for social media analysis

Ali Najafi [a], Onur Varol [a,b,*]

[a] Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey
[b] Center of Excellence in Data Analytics, Sabanci University, Istanbul, Turkey

ARTICLE INFO

Keywords:
TurkishBERTweet
Sentiment analysis
HateSpeech detection
ChatGPT

ABSTRACT

Turkish is one of the most spoken languages in the world; however, it is still among the low-resource languages. Wide us of this language on social media platforms such as Twitter, Instagram, or Tiktok and strategic position of the country in the world politics makes it appealing for the social network researchers and industry. To address this need, we introduce `TurkishBERTweet`, the first large scale pre-trained language model for Turkish social media built using over 894 million Turkish tweets. The model shares the same architecture as RoBERTa-base model with smaller input length, making `TurkishBERTweet` lighter than the most used model, called BERTurk, and can have significantly lower inference time. We trained our model using the same approach for RoBERTa model and evaluated on two tasks: Sentiment Classification and Hate Speech Detection. We demonstrate that `TurkishBERTweet` outperforms the other available alternatives on generalizability and its lower inference time gives significant advantage to process large-scale datasets. We also show custom preprocessors for social media can acquire information from platform specific entities. We also conduct comparison with the commercial solutions like OpenAI and Gemini, and other available Turkish LLMs in terms of cost and performance to demonstrate `TurkishBERTweet` is scalable and cost-effective.

1. Introduction

Social media platforms such as Twitter/X have become the primary outlet for individuals to share their opinions on various issues and react to content created by others. Increasing use of social media presents an exciting opportunity for researchers to identify trends and analyze online communities shaped by real-world events or activities of groups organized for common cause (Bas, Ogan, & Varol, 2022; Harlow, 2012; Ogan & Varol, 2017; Seckin, Atalay, Otenen, Duygu, & Varol, 2024; Segerberg & Bennett, 2011). However, the informal and concise nature of social media posts can pose challenges for analysis since most models to study textual data were trained on formal documents (Baldwin, Cook, Lui, MacKinlay, & Wang, 2013; Farzindar, Inkpen, & Hirst, 2015). Furthermore, the global nature of these platforms introduces an additional layer of complexity with multiple languages being utilized and the new concepts emerging in these dynamic social spheres.

The recent advances in natural language processing (NLP) let researchers to investigate social media platforms and they study these platforms by performing tasks like sentiment detection, topic modeling, and stance detection more accurately and consistently than traditional approaches. There has been a significant improvement in various NLP tasks with the introduction of BERT (Devlin, Chang, Lee, & Toutanova, 2019), whose structure is based on the Transformers model (Vaswani et al., 2017). Liu et al. demonstrated with RoBERTa model that BERT approach was under-trained and masked-language modeling would suffice to capture the bidirectional representations of the input (Liu et al., 2019). They also utilize Byte-Pair Encoding (BPE) (Sennrich, Haddow, & Birch, 2015) to encode input texts, which allows the model to learn representation for sub-words, mitigating the out-of-vocabulary (OOV) problem when using the models in an out-of-distribution context. Later, different variants of BERT were introduced to address the need on domain specific datasets. BERTweet model by Nguyen et al. is an example of these variants, completely trained on English Twitter datasets (Nguyen, Vu, & Nguyen, 2020).

According to ethnologue,[1] at least 85 million people speak and write Turkish, and Turkish is among the top 20 living languages in the world. In 2020, Turkish was ranked as the 11th most used language on the Twitter (Alshaabi et al., 2021), highlighting the importance of research on this widely used language. However, it is one of the low-resource languages that lacks annotated datasets for different tasks in NLP (Alecakir, Bölücü, & Can, 2022). The language models that

* Corresponding author at: Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey.
E-mail addresses: ali.najafi@sabanciuniv.edu (A. Najafi), onur.varol@sabanciuniv.edu (O. Varol).

[1] https://www.ethnologue.com/country/TR/.

https://doi.org/10.1016/j.eswa.2024.124737
Received 11 December 2023; Received in revised form 24 June 2024; Accepted 5 July 2024
Available online 10 July 2024

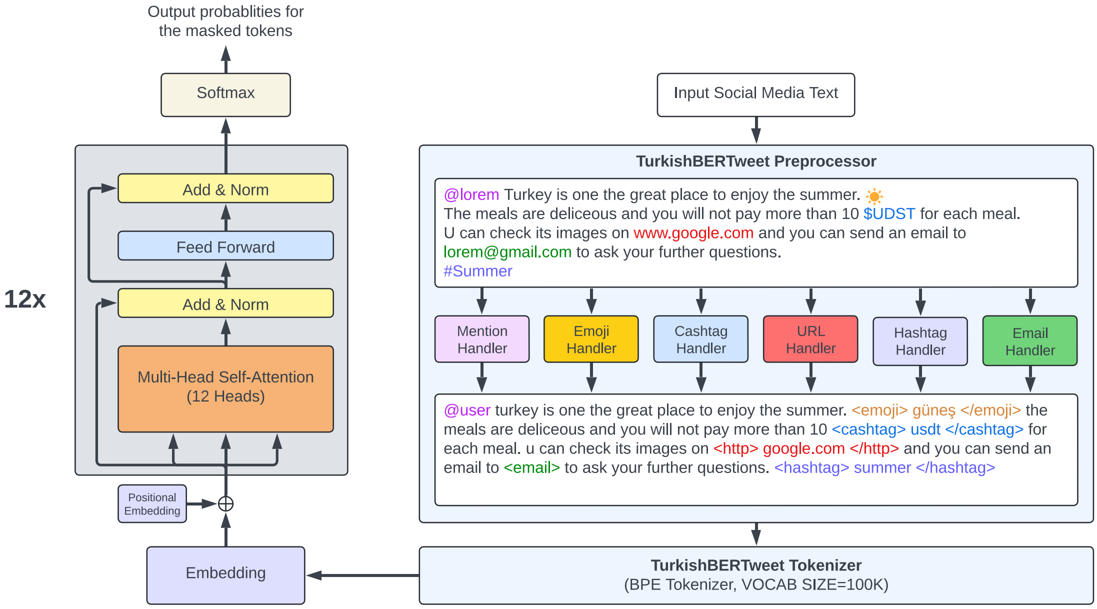

Output probablities for the masked tokens
Softmax
12x
Add & Norm
Feed Forward
Add & Norm
Multi-Head Self-Attention (12 Heads)
Positional Embedding
Embedding
Input Social Media Text
TurkishBERTweet Preprocessor
@lorem Turkey is one the great place to enjoy the summer. ☀️
The meals are deliceous and you will not pay more than 10 $UDST for each meal.
U can check its images on www.google.com and you can send an email to
lorem@gmail.com to ask your further questions.
#Summer
Mention Handler
Emoji Handler
Cashtag Handler
URL Handler
Hashtag Handler
Email Handler
@user turkey is one the great place to enjoy the summer. <emoji> güneş </emoji> the
meals are deliceous and you will not pay more than 10 <cashtag> usdt </cashtag> for
each meal. u can check its images on <http> google.com </http> and you can send an
email to <email> to ask your further questions. <hashtag> summer </hashtag>
TurkishBERTweet Tokenizer
(BPE Tokenizer, VOCAB SIZE=100K)

Fig. 1. Model Architecture. TurkishBERTweet model designed to analyze social media posts. As a first step, social media specific entities are identified and replaced with special tokens. Later the preprocessed text processed by our pre-trained tokenized which utilize byte-pair encoding to map tweets into a vocabulary of size 100 thousands. TurkishBERTweet uses 12 multi-head self-attention and use encoder blocks 12 times to provide probabilities for the mask tokens as output.

have been developed for Turkish alone are few as it is one of the low-resource languages (Alecakir et al., 2022). Models trained with multilingual data also perform better on languages with more training data or data gathering steps for such models tend to have more data quality issues on low-resource languages. The BERTurk model by Schweter (2020), which is trained on Turkish OSCAR corpus and Wikipedia Dump, is the most popular model that has been employed vastly by Turkish NLP community for wide range of tasks. Recently Kesgin et al. presented results on transformer-based models trained and evaluated with different model sizes on downstream tasks; however, their contributions were not specifically on a specific domains like social media (Toprak Kesgin, Yuce, & Amasyali, 2023). Recently, foundational models like LLama-3 (AI@Meta, 2024; Touvron et al., 2023) became available open source. These large language models (LLMs) are trained on massive multilingual datasets using significant resources and compute power.

In this work, we introduce TurkishBERTweet, a pre-trained model on Turkish Twitter dataset that contains over 894M tweets spanning 10 years of online activities between 2010 and 2020 to specifically capture the nuanced language used on social media platforms. The TurkishBERTweet model is developed for researchers who tackle social media analysis tasks since these platforms contain informal language with irregular vocabularies. Combining TurkishBERTweet model and publicly available social media datasets like #Secim2023 contributed by our team (Najafi et al., 2024), research community can conduct interdisciplinary research and pursue important societal questions using online data. We also hope that the Turkish NLP community adopts this model as a strong baseline for their further studies. We made the following contributions by developing the TurkishBERTweet model:

- We introduce the first large-scale pre-trained language model built on a rich collection of Turkish tweets. We compare this model against different existing models, multi-lingual models, fine-tuned ChatGPT models, `LLama-2-7b-chat-hf`, `Llama-3-8B-Instruct`, `Gemini 1.0 Pro`, and other available Turkish LLM models. These benchmarks on two different task across 8 datasets pose a great comparison of model performances.
- Our experimental results yield comparable performance (within 1% difference of F1 score) to larger pre-trained model (BERTurk) and achieves significantly better results than strong baselines (mBERT and TurkishAlbert) when models are evaluated within same datasets.
- Generalizability of TurkishBERTweet model shows superior performance when we experiment with leave-one-dataset-out tasks. The performance increase compared to second best model can get as high as 16% or 0.08 point increase in F1-score on the Hate Speech Detection Task.
- We introduce custom preprocessors for social media specific entities such as emojis, hashtags, mentions, and cashtags. We demonstrated the representations learned for those entities are useful for different task by presenting two case studies.
- Beyond predictive performance, inference time and cost of collecting results from models are other crucial parameters for large-scale projects and real-time analysis. Our experiments show that TurkishBERTweet's inference time is the best compared all other models and it can run on an accessible commercial hardware.
- We made our model TurkishBERTweet and its LoRA adaptors for sentiment and hate speech detection tasks publicly accessible on Huggingface platform which can be used with *transformers* library (Wolf et al., 2020). The codes and experimental results are available on Github.

2. TurkishBERTweet

In this section, we describe (i) the architecture of our model, (ii) Turkish Twitter dataset incorporated for pre-training, (iii) the special tokenizer we developed for social media analysis, and (iv) optimization model training details.

2.1. Architecture

The architecture of our model follows the structure of the RoBERTa$_{\text{base}}$ model (Liu et al., 2019). Instead of using input length as 512, we select 128 as input length for our model considering the short texts of social media. This modification makes our model approximately 21.5M parameters smaller than the BERTurk language model, which mimics RoBERTa$_{\text{base}}$. For implementation of the model, we use the Flax/Jax library provided by the Transformers package. As Fig. 1 illustrates, our model has 12 layers. Each block uses 12 self-attention heads with a hidden dimension of 768.

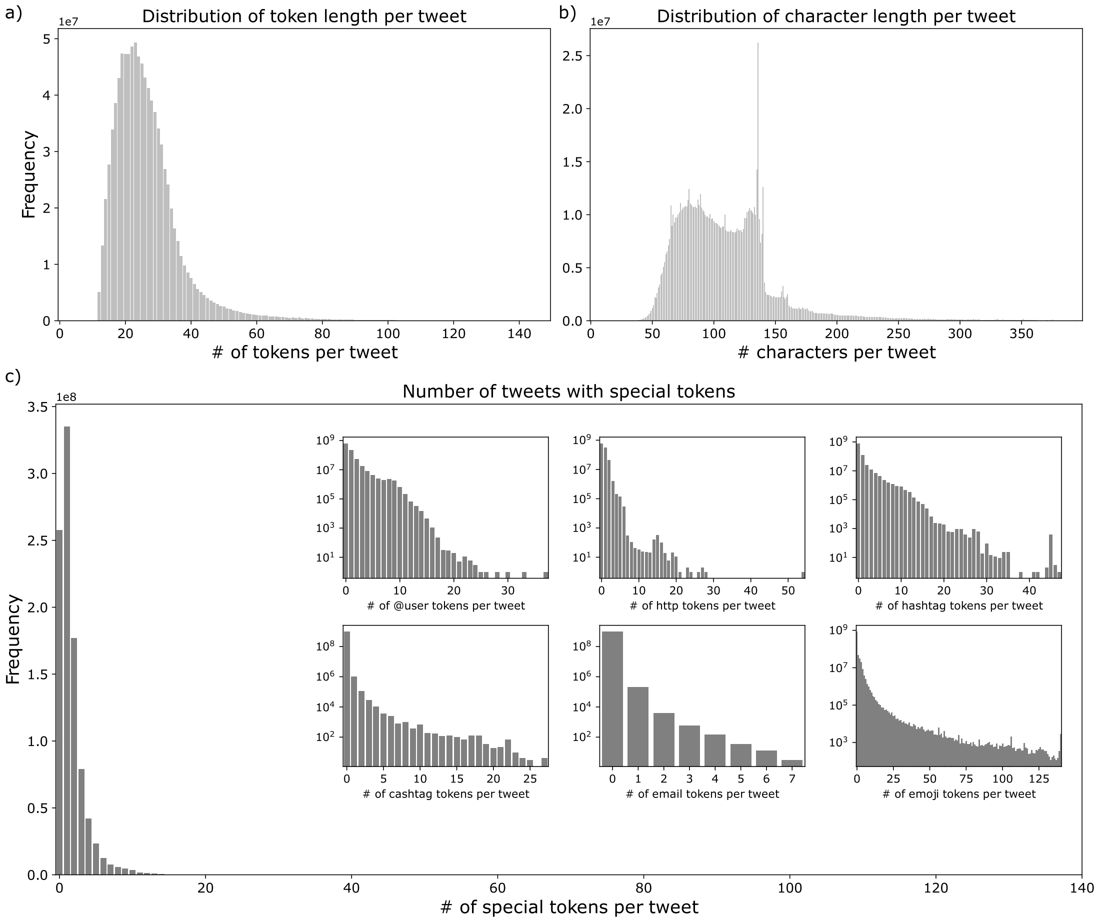


Fig. 2. Descriptive statistics of Turkish social media corpora. Social media text tend to be short as the number of tokens (a) and characters (b) per tweet presented in histograms. Despite the short length of posts, special entities can convey valuable information and distributions of these entities presented for all tweets and per entity (c).

2.2. *Pre-training data*

The dataset captures over 10 years of online activity capturing Turkish Twitter activities between 2010 and 2020. Since the dataset acquired through the Twitter Streaming API, we could collect content covering various important social events and daily discussions. Considering the important social events occurring in Türkiye in the past 10 years, this dataset reflects the online discussions about 6 elections, early-phases of COVID-19 pandemic, 2016 coup attempt, and various other important political and social events.

Since the data stream also captures retweeted content, we filtered the retweeted posts and retained only the original content posted on the platform. We also exclude tweets that contains only single entities and tweets with fewer than 10 tokens. Our final Turkish pre-training dataset includes 110 GB of uncompressed text with nearly 894 million tweets. Our dataset presents the characteristics of social media posts where few social media accounts responsible with creation of several content and most of the content produced to communicate with other platform users tend to be short texts.

2.3. *Tokenizer*

Since social media posts contain specialized entities, we defined additional tokens to capture them in the text. We added extra special tokens such as `@user`, `<hashtag>`, `</hashtag>`, `<cashtag>`, `</cashtag>`, `<emoji>`, `</emoji>`, `<http>`, and `</http>`. The closing tags define the boundaries of special tokens, which are used for unmasking entities such as emojis, hashtags, etc. Including these special tokens in the tokenizer allows us to extract information from the tweets without the need for direct supervised learning.

We trained a fastBPE tokenizer (Sennrich, Haddow, & Birch, 2016) with a vocabulary size of 100 thousand. Before feeding the data into the model, we applied the preprocessing steps listed in the Fig. 1. We used the `emoji`[2] package to replace emojis with their equivalent texts, although Turkish equivalents were not available for all emojis. To address this issue, we translated them into Turkish using Google Translator. Additionally, the domains of URL links were extracted and included in the http tokens. Moreover, the **@user** token was used in place of mentions and emails, which are denoted by the <**email**> token in tweets, to ensure privacy. By detecting cash signs in tweets, we encapsulated them with the **cashtag** token. Fig. 2(a,b) shows the distribution of tokens and characters per tweet after the preprocessing steps. Fig. 2(c) and its inset figures present the distributions of all special tokens per tweet.

2.4. *Optimization*

We use the RoBERTa implementation from the `transformers` package of Huggingface and initialized the model with random weights. We set the maximum input length to 128 for the model. For optimizing the model, we followed Liu et al. (2019) and used Adam optimizer (Kingma & Ba, 2014) with a batch size of 128 per TPU pod, totaling $8 * 128 = 1024$ using all available TPU pods provided by Google Cloud Research. We trained the model for seven days, achieving a peak learning rate of 1e−5.

[2] https://pypi.org/project/emoji.

Table 1
Sentiment detection datasets. Descriptive statistics of the datasets and their class distributions for three categories presented.

Dataset	# of instance	Positive	Neutral	Negative
VRLSentiment	23,689	5,469	10,146	8,074
TSATweets[a]	6,001	1,552	1,448	3,001
Kemik-17bin[b] (Amasyali, Tasköprü, & Çaliskan, 2018)	17,289	4,579	5,822	6,888
Kemik-3000[b] (Çetin & Amasyalı, 2013)	3,000	756	957	1,287
BOUN (Köksal & Özgür, 2021)	4,733	1,271	2,769	693
TSAD[c]	489,644	262,166	170,917	56,561

[a] https://github.com/sercankulcu/sentiment-analysis-of-tweets-in-Turkish.

[b] http://www.kemik.yildiz.edu.tr/veri_kumelerimiz.html.

[c] https://huggingface.co/datasets/winvoker/turkish-sentiment-analysis-dataset.

3. Experimental setup

To present comprehensive experiment and detailed evaluation of TurkishBERTweet model, we focused on two downstream tasks: sentiment and hate speech detection. Performance of our model compared against state-of-the-art models and publicly available large language models. Models were also fine-tuned for these tasks following standard and LoRA fine-tuning. Other pre-trained LLMs were evaluated with zero-shot scenarios.

3.1. Datasets for downstream tasks

As mentioned earlier, Turkish is one of the low-resource languages for which there are not many annotated datasets available. With this in mind, we evaluated the models on two text classification tasks where reliable and sufficient data could be found: Sentiment Analysis and Hate Speech detection. To quantify the consistency and generalizability of the models on novel datasets, we measured their performance not only in a cross-validated setting but also in experiments with out-of-dataset configurations.

3.1.1. Sentiment analysis

We evaluate the models on the Sentiment Analysis datasets as shown in Table 1. In the Turkish NLP community, almost all available sentiment analysis models are trained as binary classification models, meaning that an input is either positive or negative, which is not always the case since a text can also have a neutral sentiment if the discussion is not polarized or is simply stating factual information. To fill this gap, we provide our final sentiment detection model as a three-class classifier.

We searched for different publicly available and manually labeled tweet datasets for our experiments. Some datasets provide unique identifiers of tweets; however, the majority of these tweets were either removed or posted by deleted accounts. The VRLSentiment dataset contains political tweets annotated by students as part of a research project in our group. We found the TSATweets on a GitHub repository, and the Kemik datasets were requested from a researcher via email. The BOUN dataset mostly contains tweets commenting about universities in Türkiye, which means that it covers only a narrow distribution of the Twitter platform. The TSAD dataset differs from other datasets as it captures product reviews and Turkish Wikipedia entries.

3.1.2. Hate speech detection

Table 2
Hate speech detection datasets. The distribution of classes for two datasets. HateSpeech SIU dataset also provides left-out evaluation set as test set. HSD2LANG released only one dataset and kept evaluation set as private.

Dataset	Class	Train set	Test set
HateSpeech SIU	No Hate speech	3493	873
	Hate speech	1190	298
	Total	**4683**	**1171**
HSD2LANG	No Hate speech	6121	NA
	Hate speech	2684	NA
	Total	**8805**	NA

We test our model on two hate speech datasets. The first dataset was created as part of the Computational Social Sciences Session of the 2023 Signal Processing and Communication Applications Conference (SIU) (Arın et al., 2023). Organizers released the tweet IDs and their corresponding hate speech classifications for the competition. We rehydrated all tweets accessible at the time the dataset was released for the competition. We experimented with the train/test split provided for the evaluation to compare our model against the leaderboard. In addition, we performed a 10-fold cross-validation experiment by combining the training and test sets. The second dataset, HSD2LANG, was obtained from an ACL workshop competition as part of the EACL'2024 conference (Uludoğan, Dehghan et al., 2024). This dataset is prepared for hate speech detection task about refugees, the Israel–Palestine conflict, and anti-Greek discourse. Table 2 shows the distribution of labels in these two datasets for binary classification. It is important to mention that these two datasets have 2,311 overlapping samples. The results presented in Section 4.2 for out-of-distribution analysis, we removed these samples from the HSD2LANG dataset, resulting in the dataset containing 1995 and 4499 samples for content with and without hate speech, respectively.

3.2. Baselines models for benchmark

We compare our model with various language models that have different base architectures and are widely used across different fields. These language models that we experimented with are listed below and we refer to their academic publications and code repositories when available. The set of Large Language Models used in zero-shot experiments were also selected by using a public OpenLLM Turkish leaderboard.[3] **BERTurk**[4] is a well-known language model within the Turkish NLP community and has been widely used. This model is trained on 35 GB of Turkish text data and has a vocabulary of 128 thousands tokens. This model is available in different versions. According to the model card on the HuggingFace platform, it was trained using a collection from the OSCAR corpus, a Wikipedia dump, and various OPUS corpora (Schweter, 2020). The OSCAR dataset includes 5,000 tweets (Çarık & Yeniterzi, 2022), indicating that the model has been exposed to social media text (Abadji, Ortiz Suarez, Romary, & Sagot, 2022; Abadji, Suárez, Romary, & Sagot, 2021; Caswell et al., 2021; Ortiz Su'arez, Romary, & Sagot, 2020; Ortiz Su'arez, Sagot, & Romary, 2019).

mBERT[5] is trained with content from the largest 104 languages on Wikipedia. It utilizes a word piece tokenizer and sets the vocabulary size to 110 thousands. Languages with more Wikipedia pages were under-sampled, while those with fewer pages were over-sampled to create a balanced input dataset. Unfortunately, no further information

[3] https://huggingface.co/spaces/malhajar/OpenLLMTurkishLeaderboard.

[4] https://huggingface.co/dbmdz/bert-base-turkish-128k-uncased.

[5] https://huggingface.co/bert-base-multilingual-cased.

is provided regarding the proportion of languages (Devlin, Chang, Lee, & Toutanova, 2018).

ConvBERTurk is the Turkish version of ConvBERT model (Jiang et al., 2020). We obtained the model `convbert-base-turkish-cased`[6] for our experiments from HuggingFace platform. ConvBERT models utilize a convolutional kernel to capture local similarities between tokens. These similarities are then incorporated into self-attention to create a mixed attention block.

TurkishAlbert[7] model contains almost 12M parameters, making it smaller than all other models. It was trained on 200 GB of Turkish text, which was collected from various sources including online blogs, free e-books, newspapers, the Common Crawl corpus, Twitter, articles, and Wikipedia. The tokenizer for this model has a vocabulary size of 32k. This model is one of the variants of Albert model proposed by Lan et al. (2019).

mT5-Large[8] is the multilingual version of the T5 language model introduced by Raffel et al.. This model was trained on mC4 datasets that contains almost 71B Turkish tokens and Turkish texts accounts for 1.93% of their training dataset (Xue et al., 2020).

TURNA[9] is an encoder–decoder model trained on multiple Turkish datasets, predominantly on the mC4 and OSCAR datasets, and was trained on 42.7 billion tokens (Uludoğan, Balal et al., 2024). As an encoder–decoder model, we fine-tuned it in a sequence-to-sequence setting.

Llama-3-70B-Instruct[10] and **Llama-3-8B-Instruct**[11] are 70B and 8B versions of Meta's Llama models which have been instruction fine-tuned. They support Turkish and they are capable of generating Turkish texts (AI@Meta, 2024).

Llama-2-7b-chat-hf[12] model (Touvron et al., 2023) was not trained on any Turkish text during its pre-training phase; instead, the majority of its corpus comprises English texts. Nevertheless, as a foundation model, it presents an opportunity for fine-tuning to assess its performance on Turkish texts. We utilized the 7B version of LLama-2 in a zero-shot setting to evaluate its performance.

Trendyol-LLM-7b-chat-dpo-v1.0[13] is based on Mistral 7B (Jiang et al., 2023) large language model that uses an optimized transformer architecture. This model is DPO fine-tuned (Rafailov et al., 2024) on 11K sets of prompt-chosen-reject samples.

Turkcell-LLM-7b-v1[14] is an extended version of a Mistral 7B (Jiang et al., 2023) Large Language Model for Turkish. It was trained on a cleaned Turkish raw dataset containing 5 billion tokens. The training process involved using the DoRA (Liu et al., 2024) method initially and they utilized Turkish instruction sets created from various open-source and internal resources for fine-tuning with the LORA method.

Orbina/Orbita-v0.1[15]is a Qwen-based large language model (Bai et al., 2023), but unfortunately there is no clear information regrading its pretrained data. It has 14B parameters and fully finetuned on Turkish texts.

GPT-4o and **GPT3.5-turbo** are proprietary models by OpenAI. Unfortunately, there is no confirmed public information at the time of this publication about the training dataset or the pipelines used by OpenAI to prepare these models (OpenAI, 2023). We used the paid API from OpenAI to fine-tune models with our own datasets and collect responses for our prompts.

Gemini 1.0 Pro is one of the variants of the Gemini models developed by Google (Team et al., 2023). Similar to OpenAI's models, there is no information available regarding their pre-trained datasets or their training pipelines.

3.3. Fine-tuning pre-trained language models

The fine-tuning procedure uses a pre-trained language model and adapts it for use in a specific task. There are different approaches introduced in the literature to build task-specific models by reducing computational cost as much as possible. In this work, we experiment with the full fine-tuning and low-rank adaptation (LoRA) fine-tuning methods (Hu et al., 2021) to compare and evaluate the models for downstream tasks. To ensure comparable results, we performed 10-fold stratified cross-validation to preserve the proportions of the classes in training and testing and to maintain consistent performance across each dataset. We implemented two different fine-tuning approaches; however, we only conducted experiments with LoRA fine-tuning for the models that performed best in the standard fine-tuning experiments since the performance of LoRA fine-tuned models are superiors to standard fine-tuning. Additionally, we investigated the performance of generative models in a zero-shot setting by creating a prompt for the two tasks that we are evaluating.

Full Fine-tuning (FT): In this approach, all or some of the original parameters of the model are updated based on a given dataset. Using this method, we compare our LLM with the baselines by freezing all models' parameters except for adding a final pooling layer followed by a dense classification layer. Then, the models were trained for 50 epochs per fold, selecting the best model for the final evaluation of each fold. Early stopping was used to prevent overfitting.

LoRA Fine-tuning (LFT): LoRA is a low-rank adaptation technique for large language models proposed by Hu et al. (2021). It operates by freezing the pre-trained weights and injecting trainable rank decomposition matrices into each layer of the Transformer architecture. This method reduces the number of trainable parameters while preserving the knowledge learned from the pre-trained model, thus enabling more efficient fine-tuning of large language models for downstream tasks. Using the PEFT library (Mangrulkar, Gugger, Debut, Belkada, & Paul, 2022) provided by HuggingFace, the models were trained for ten epochs with a rank of $r = 8$ and a scaling parameter of $\alpha = 16$. For encoder models, the `query` and `value` modules were specifically targeted with the sequence classification objective. Fig. 3 illustrates the instruction/prompt structures used for fine-tuning the generative models. For the TURNA and mT5-Large models, with the same rank and alpha, we fine-tuned these models with a context size of 512 on the Sequence to Sequence Modeling (Seq2Seq) objective. We quantized these models in 4-bits, and since they were exposed to Turkish data in their pre-trained datasets, we used Turkish instructions. We prepared the data as suggested by OpenAI's pipelines for fine-tuning GPT-3.5 Turbo, providing contents for three roles of a chatbot: `system`, `assistant`, and `content`. We trained GPT3.5 Turbo on out-of-distribution datasets for one epoch.

Zero-Shot (ZS): We investigated the performance of `Llama-2-7b-chat`, `Llama-3-8B-Instruct`, `Llama-3-70B-Instruct`, `Trendyol-LLM-7b-chatdpo-v1.0`, `Turkcell-LLM-7b-v1`, and `Orbina/Orbita-v0.1`. We quantized the `Llama-3-70B-Instruct` model due to its size. All of these models are optimized for dialogue and chat use cases, which enables us to evaluate their performance on downstream tasks. Additionally, we also used `Gemini 1.0 Pro` and two versions of ChatGPT models, namely GPT-4o and GPT-3.5-turbo, in our experiments to assess their performance. We collected inferences from these models by simply prompting them using the prompt structure illustrated in Fig. 3. The response text can sometimes contain additional text or English answers, so we are post-processing the responses to create final output label.

[6] https://huggingface.co/dbmdz/convbert-base-turkish-cased.

[7] https://huggingface.co/loodos/albert-base-turkish-uncased.

[8] https://huggingface.co/google/mt5-large.

[9] https://huggingface.co/boun-tabi-LMG/TURNA.

[10] https://huggingface.co/meta-llama/Meta-Llama-3-70B-Instruct.

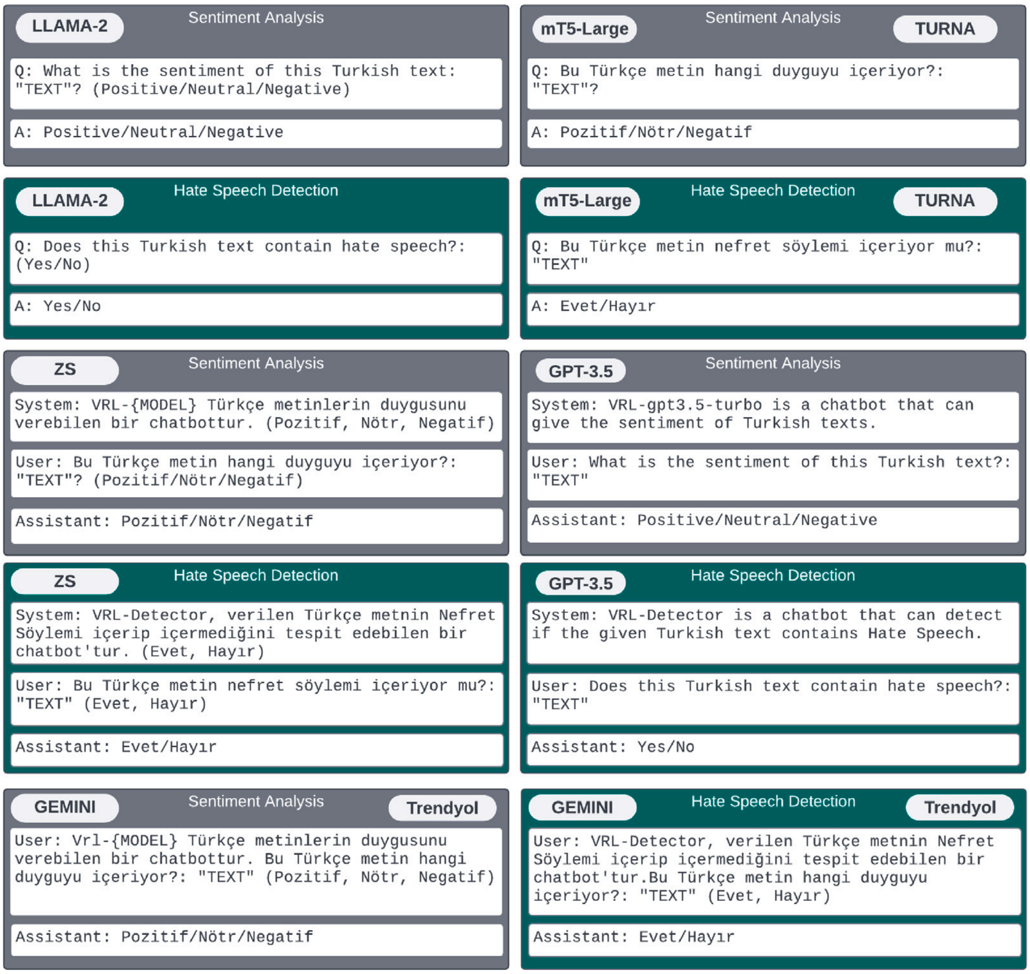
[11] https://huggingface.co/meta-llama/Meta-LLama-3-8B-Instruct.

[12] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf.

[13] https://huggingface.co/Trendyol/Trendyol-LLM-7b-chat-dpo-v1.0.

[14] https://huggingface.co/TURKCELL/Turkcell-LLM-7b-v1.

[15] https://huggingface.co/Orbina/Orbita-v0.1.



LLAMA-2 — Sentiment Analysis

Q: What is the sentiment of this Turkish text: "TEXT"? (Positive/Neutral/Negative)

A: Positive/Neutral/Negative

mT5-Large — Sentiment Analysis — TURNA

Q: Bu Türkçe metin hangi duyguyu içeriyor?: "TEXT"?

A: Pozitif/Nötr/Negatif

LLAMA-2 — Hate Speech Detection

Q: Does this Turkish text contain hate speech?: (Yes/No)

A: Yes/No

mT5-Large — Hate Speech Detection — TURNA

Q: Bu Türkçe metin nefret söylemi içeriyor mu?: "TEXT"

A: Evet/Hayır

ZS — Sentiment Analysis

System: VRL-{MODEL} Türkçe metinlerin duygusunu verebilen bir chatbottur. (Pozitif, Nötr, Negatif)

User: Bu Türkçe metin hangi duyguyu içeriyor?: "TEXT"? (Pozitif/Nötr/Negatif)

Assistant: Pozitif/Nötr/Negatif

GPT-3.5 — Sentiment Analysis

System: VRL-gpt3.5-turbo is a chatbot that can give the sentiment of Turkish texts.

User: What is the sentiment of this Turkish text?: "TEXT"

Assistant: Positive/Neutral/Negative

ZS — Hate Speech Detection

System: VRL-Detector, verilen Türkçe metnin Nefret Söylemi içerip içermediğini tespit edebilen bir chatbot'tur. (Evet, Hayır)

User: Bu Türkçe metin nefret söylemi içeriyor mu?: "TEXT" (Evet, Hayır)

Assistant: Evet/Hayır

GPT-3.5 — Hate Speech Detection

System: VRL-Detector is a chatbot that can detect if the given Turkish text contains Hate Speech.

User: Does this Turkish text contain hate speech?: "TEXT"

Assistant: Yes/No

GEMINI — Sentiment Analysis — Trendyol

User: Vrl-{MODEL} Türkçe metinlerin duygusunu verebilen bir chatbottur. Bu Türkçe metin hangi duyguyu içeriyor?: "TEXT" (Pozitif, Nötr, Negatif)

Assistant: Pozitif/Nötr/Negatif

GEMINI — Hate Speech Detection — Trendyol

User: VRL-Detector, verilen Türkçe metnin Nefret Söylemi içerip içermediğini tespit edebilen bir chatbot'tur.Bu Türkçe metin hangi duyguyu içeriyor?: "TEXT" (Evet, Hayır)

Assistant: Evet/Hayır

Fig. 3. Prompt Structures. This figure shows the prompt structures that we used in prompting and instruction fine-tuning the generative models for Sentiment Analysis and Hate Speech Detection tasks.

4. Experimental results

To compare TurkishBERTweet with other available models, we conduct a series of experiments on different datasets we introduced earlier, and we use 10-fold cross-validation for each task. Table 3 presents the results obtained for sentiment and hate speech detection tasks.

4.1. Model comparisons

We observed significant improvements in both tasks and across various datasets when fine-tuning was applied with the LoRA method during training. The two most successful models, BERTurk and TurkishBERTweet, demonstrated comparable performance across different datasets for sentiment analysis tasks. Since most applications of BERTurk employ a standard fine-tuning approach, our publicly available model (the TurkishBERTweet model with LoRA) is much more preferable and achieves 4%–9% higher performance than the BERTurk model with standard fine-tuning. TurkishAlbert and mBERT models perform the least in all settings, which can be due to lack of Turkish data used in mBERT training and number of parameters in the model. We also experimented with TURNA and mT5-Large to compare TurkishBERTweet with models considerably larger. Despite their size, these model performed beyond TurkishBERTweet and BERTurk on all datasets.

For Hate Speech detection, like sentiment analysis, we performed 10-fold cross-validation to evaluate the performance of the models. We also used the training and testing splits from the SIU 2023 hate speech detection competition. Using the dataset provided in the competition, we obtained a macro-F1 score of 0.73167 for TurkishBERTweet with LoRA fine-tuning, which is higher than the submission top-ranked in the competition with its score of 0.72167. These scores are reported on the contest page on Kaggle.[16] For HSD2LANG dataset, in a similar setting, we see slight performance increase for TurkishBERTweet compared to BERTurk. Need to mention that in the competition held by EACL2024 workshop, TurkishBERTweet gained higher private score that shows a better generalization compared to BERTurk (Najafi & Varol, 2024) and gained the 2nd and 3rd ranks in this competition. The team that ranked 2nd used our public model on HuggingFace and fine-tuned it better for the task than our teams submission which ranked 3rd. Although, the team that won the 1st rank use ConvBERTurk, their training dataset was augmented by translating Arabic texts to Turkish (Uludoğan, Dehghan et al., 2024).

Our experiments also contains several LLMs that have multilingual capabilities such as Llama, ChatGPT, and Gemini; as well as, models fine-tuned with Turkish tasks and introduced by different industry research teams. We use these models in zero-shot setting and the results are presented in Table 3. Among these 8 different models, ChatGPT4o performs the best in all datasets with 0.04 to 0.06 higher F1-score; however, the performance is still behind nearly 2%–8% for most datasets when compared to performance of the best fine-tuned model. These models also seems unable to perform on hate speech

[16] https://www.kaggle.com/competitions/siu2023-nst-task2.

Table 3
Weighted F1-score of the baseline models for sentiment and hate speech tasks. We evaluated different settings like LoRA finetuning (LFT) and standard fine tuning (FT), as well as zero-shot (ZS) evaluation through prompts. Best scores are presented in bold font and when the difference is not significant more than one model highlighted.

Task	Model	VRLSentiment	Kemik-17bin	Kemik-3000	TSATweets	BOUN	TSAD	HateSpeech SIU	HSD2Lang
LFT	TurkishBERTweet	**0.642 ± 0.008**	**0.758 ± 0.011**	**0.662 ± 0.025**	**0.715 ± 0.012**	**0.730 ± 0.022**	**0.969 ± 0.001**	**0.807 ± 0.013**	**0.815 ± 0.013**
	BERTurk	**0.640 ± 0.013**	**0.778 ± 0.008**	**0.688 ± 0.031**	**0.713 ± 0.014**	**0.752 ± 0.020**	**0.973 ± 0.001**	**0.811 ± 0.012**	**0.810 ± 0.012**
	ConvBERTurk	**0.639 ± 0.012**	**0.779 ± 0.008**	**0.682 ± 0.013**	0.658 ± 0.013	0.696 ± 0.021	**0.975 ± 0.001**	**0.814 ± 0.012**	**0.813 ± 0.013**
	mBERT	0.579 ± 0.008	0.686 ± 0.001	0.536 ± 0.020	0.637 ± 0.017	**0.752 ± 0.012**	0.959 ± 0.011	0.740 ± 0.037	0.787 ± 0.011
	TurkishAlbert	0.595 ± 0.010	0.680 ± 0.010	0.596 ± 0.028	0.645 ± 0.013	0.698 ± 0.019	0.897 ± 0.001	0.759 ± 0.018	0.778 ± 0.011
	TURNA	0.622 ± 0.012	0.482 ± 0.047	0.505 ± 0.051	0.595 ± 0.018	0.627 ± 0.031	NA	0.778 ± 0.017	0.818 ± 0.013
	mt5-Large	0.629 ± 0.010	0.750 ± 0.015	0.485 ± 0.063	0.613 ± 0.066	0.709 ± 0.021	NA	0.775 ± 0.015	0.807 ± 0.011
FT	TurkishBERTweet	0.613 ± 0.012	0.703 ± 0.008	0.621 ± 0.027	0.670 ± 0.011	0.690 ± 0.029	0.915 ± 0.001	0.753 ± 0.015	0.764 ± 0.015
	BERTurk	0.590 ± 0.008	0.701 ± 0.011	0.634 ± 0.023	0.655 ± 0.016	0.729 ± 0.021	0.937 ± 0.001	0.752 ± 0.011	0.764 ± 0.010
	ConvBERTurk	0.561 ± 0.009	0.637 ± 0.011	0.632 ± 0.014	0.658 ± 0.013	0.696 ± 0.021	0.942 ± 0.001	0.713 ± 0.023	0.739 ± 0.017
	mBERT	0.537 ± 0.005	0.598 ± 0.014	0.523 ± 0.028	0.598 ± 0.012	0.659 ± 0.029	0.883 ± 0.001	0.715 ± 0.018	0.725 ± 0.013
	TurkishAlbert	0.545 ± 0.010	0.637 ± 0.011	0.580 ± 0.033	0.603 ± 0.015	0.676 ± 0.021	0.897 ± 0.001	0.725 ± 0.018	0.715 ± 0.014
ZS	Llama-3-70B-Instruct	0.562	0.625	0.592	0.653	0.578	NA	0.355	0.392
	Llama-3-8B-Instruct	0.406	0.500	0.477	0.580	0.310	NA	0.187	0.224
	Llama-2-7B-chat-hf	0.437	0.454	0.458	0.455	0.434	NA	0.442	0.431
	ChatGPT4o	0.628	0.689	0.637	0.691	0.584	NA	0.504	0.587
	Gemini 1.0 Pro	0.537	0.632	0.591	0.655	0.411	NA	0.348	0.421
	Orbita-v0.1	0.463	0.485	0.489	0.567	0.321	NA	0.280	0.321
	Turkcell-LLM-7b-v1	0.431	0.493	0.459	0.527	0.383	NA	0.291	0.356
	Trendyol-LLM-7b-chat-dpo-v1.0	0.444	0.521	0.518	0.516	0.486	NA	0.296	0.381

Table 4
Weighted F1-score for leave-one-dataset-out evaluation. In each experiment, we keep a dataset (D) for evaluation while others ($\forall - \{D\}$) in the same category were used in model training.

Dataset (D)	TurkishBERTweet (LFT)	BERTurk (LFT)	TurkishBERTweet (FT)	BERTurk (FT)	GPT3.5-Turbo
VRLSentiment	0.556 ± 0.008	0.566 ± 0.008	0.547 ± 0.011	0.519 ± 0.007	0.555
Kemik–17bin	0.650 ± 0.010	0.671 ± 0.010	0.604 ± 0.011	0.618 ± 0.013	0.653
Kemik–3000	0.650 ± 0.026	0.671 ± 0.017	0.578 ± 0.031	0.595 ± 0.029	0.637
TSATweets	0.608 ± 0.008	0.631 ± 0.015	0.576 ± 0.028	0.583 ± 0.026	0.550
BOUN	0.616 ± 0.022	0.628 ± 0.024	0.610 ± 0.025	0.635 ± 0.016	0.580
HateSpeech SIU	0.840 ± 0.012	0.814 ± 0.015	0.763 ± 0.012	0.754 ± 0.013	0.808
HSD2Lang	0.781 ± 0.011	0.725 ± 0.056	0.707 ± 0.013	0.691 ± 0.020	0.785

detection task since these models tend to provide cautious responses by considering most input containing hate speech.

4.2. Out-of-domain evaluation

To investigate the generalizability of the models on different domains, we performed an out-of-distribution evaluation in which we left one of the datasets out and trained the models on the rest of the datasets. To be able to perform cross validation, we divide instances of combined training datasets and left-out dataset for testing into number of fold. This way we can train different models and make sure the testing and training instances will come from different datasets. We focused on the top performing models from Table 3, namely TurkishBERTweet with and BERTurk with standard (FT) and LoRA fine-tuning (LFT), for this analysis. Since the community uses BERTurk models with standard fine-tuning frequently, we are also reporting performance of that model as comparison.

We witnessed –not a surprising– performance decrease in some cases as much as 18% for both TurkishBERTweet and BERTurk models, since the testing datasets are different from the ones provided for training in this challenging and more realistic setting. It is worth mentioning that TurkishBERTweet (LFT) still outperforms the BERTurk (FT) language model almost on all of the datasets except BOUN and BERTurk (LFT) is achieve comparable ±0.01 performance to TurkishBERTweet. Models tested on hate speech detection tasks, TurkishBERTweet (LFT) outperforms all models that we interpret as a promising insight for generalizability.

We also fine-tuned the ChatGPT3.5 Turbo models for this experiment, and this model achieved ±0.01 scores compared to TurkishBERTweet in 5 out of 7 experiments. However, in two cases TurkishBERTweet achieve nearly 0.05 higher F1-score.

4.3. Inference time comparison

In addition to comparing models based on performance, we can also measure inference time and model sizes to consider their usability in large-scale analysis. In terms of input length of the models, TurkishBERTweet works with input length of 128, which is half of the input length for BERTurk. This property of the model reduces the size of the model significantly. Consequently, the batch size can be increased to load more data onto the GPU during inference time. To compare the inference time of the models, we created multiple sets of tweets with sample sizes ranging from 2^0 to 2^{12}. We set the batch sizes for different models to the maximum values that could be accommodated by the GPU. The batch sizes were set to 2^3, 2^6, 2^7, and 2^{11} for Llama-3-8B, TURNA, mt5-large, and the rest of the models, respectively. The purpose of having multiple sets of tweets is to monitor model performance as the sample sizes become less than, equal to, or greater than the optimal batch size. It should be noted that we padded the input texts into 128 tokens to fairly compare models, but we also achieved similar outcome when we padded the input test to the maximum input length of the models. For each model, we fed the sets of tweets into the model 100 times in one forward pass using a `1X GeForce RTX 4090 249 GB`. We report the average inference time per sample for each set and illustrate the relationship between average inference time per sample and the number of parameters in Fig. 4. As stated in Fig. 4, the inference time decreases as the set size increases and stabilizes when the sample size reaches 2^6. TurkishBERTweet exhibits the lowest inference time compared to the other models, leading 16% faster inference time to its closest competition, and more than one order of magnitude faster than models like Llama-3, TURNA, and mT5-Large.

This practical comparison points that TurkishBERTweet model is more suitable to process millions of tweets significantly faster for social

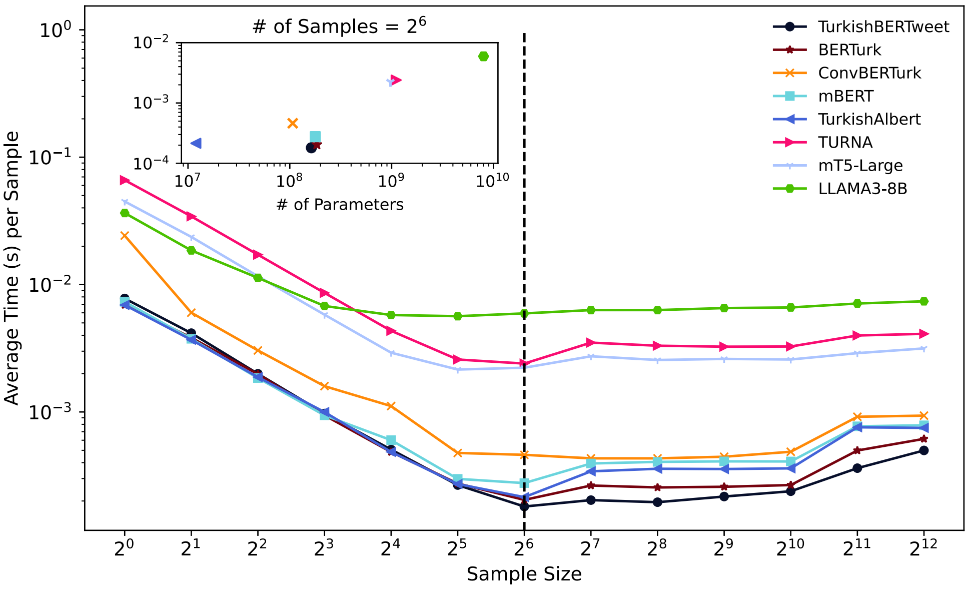


Fig. 4. Estimation of inference time per sample for different batch size and models. Average time per sample estimated over 100 repetition. Batch sizes for powers of two are considered for evaluation and different models with 128 context length in a single forward pass are tested for comparison.

media analysis. For instance, Firehose data stream (all public tweet) of Twitter produces about 4,000 public tweets per second (Pfeffer et al., 2023). Considering less than 10% of public tweets posted in Turkish, we can process such data streams in real time with TurkishBERTweet.

5. Discussion

Building a language model specifically trained on Turkish social media posts provided valuable lessons throughout the process. When we began pre-training TurkishBERTweet on the Twitter/X data, we hypothesized that a dataset composed entirely of Turkish tweets would yield improved results on downstream tasks. As demonstrated in the evaluation section, our model achieves results ±0.01 F1-score with BERTurk, except where LoRA fine-tuning led to significant performance improvements compared to other publicly available models. This finding aligns well with the conclusions of the BERTweet paper (Nguyen et al., 2020), which suggests that smaller models pre-trained on domain-specific datasets can achieve better performance. The authors reported almost a 2-point increase in F1-score for text classification and nearly identical performance in the NER task. This finding is also consistent with discussions regarding the quality of the pre-trained dataset (Longpre et al., 2023).

It is also important to mention that prompt construction is another factor in the performance of generative models, and their response quality can be improved if more context is given about the task. We did not explore this topic in depth because the prompt construction of generative models is outside our research scope. We see the very poor performance of Llama2-7b-Chat in the Zero Shot classification setting, which was predictable because the model has not seen any Turkish texts. However, the Llama-3-8B-Instruct model achieved comparative performance to our TurkishBERTweet model.

In our experiments for out-of-domain evaluation, we see a decrease in the models' performance compared to the single dataset evaluation. This outcome is expected to a certain extend since each dataset may have similar instances across training and test sets; however, different datasets can vary temporally and topically. For that reason, this experiment poses a great benchmark for evaluating generalizability of models. We also observed that the performance of GPT3.5-Turbo was almost similar to the performance of our proposed model, emphasizing that our model is more preferable since its available open-source and free of charge to use.

5.1. *Representation from preprocessors*

TurkishBERTweet model offers a custom preprocessor to process social media specific entities such as emojis, hashtags, cashtags etc. as introduced in Section 2.1. Here we demonstrate the value created by providing custom preprocessors for emoji and cashtag entities as case studies. Users of the TurkishBERTweet model can also tailor these systems for their own projects.

5.1.1. *Inferring emotions from emojis*

Social media users have been utilizing emojis as a way to convey their emotions (Derks, Fischer, & Bos, 2008; Kralj Novak, Smailović, Sluban, & Mozetič, 2015). We experimented with one of the most comprehensive datasets, called EmoTag1200 (Shoeb & de Melo, 2020), where nine human coders annotated a set of 150 popular emojis with regard to eight different emotions using a 5-point Likert scale. The dataset presents scores for *anger*, *anticipation*, *disgust*, *fear*, *joy*, *sadness*, *surprise*, and *trust* from the Wheel of Emotions by Plutchik (Plutchik & Kellerman, 1980).

To compare representations for different emojis, we set three different measurements, presented in Table 5. The first measurement (M1) focuses on how much the embeddings for emojis resemble the emotion scores provided by human annotators. To achieve that, we calculate the cosine similarity between the vector representation of an emoji (V_{Emoji}) and one of the emotion words (V_{w_E}). The Turkish emotion words that we used to extract the vector presentations for emojis are kızgınlık (anger), korku (fear), beklenti (anticipation), sürpriz (surprise), sevinç (joy), üzüntü (sadness), güven (trust), and iğrenme (disgust). We assume that the similarity between two vectors will correlate with the emotion scores (E) from the EmoTag1200 dataset.

Spearman's correlation between the vector similarity and emotion scores shows a stronger association for TurkishBERTweet than for BERTurk in 5 out of 8 emotions. Since some emojis may not directly relate to these emotions, we analyzed the polarization of emotions following Plutchik's theory. The measurements we have in M3 directly quantify the correlation between two opposite emotions, such as *anger* and *fear*. Although the theory suggests that these pairs of emotions should be anti-correlated — meaning that a higher annotation for one should correspond to a lower annotation for the opposite emotion — our experiment suggests that some of these emotion pairs are not direct opposites, as reflected in their positive correlation scores. For

Table 5
Inferring emotion strength from emojis. Comparing models using manually annotated emotions of emojis. M1 and M2 measure Spearman's correlation between vector similarities and emotion scores. M3 presents validation of emotion scores from EmoTag1200 dataset by measuring correlation between opposite emotions.

M1: Cosine similarity of emoji to word ($S_1 = Cos(V_{Emoji}, V_{w_E})$) vs. Emotion score (E)								
corr(S_1, E)	Anger	Fear	Anticipation	Surprise	Joy	Sadness	Trust	Disgust
TurkishBERTweet	0.58	0.61	−0.04	0.30	−0.37	0.57	−0.37	0.69
BERTurk	0.17	0.34	−0.03	−0.08	−0.15	0.46	−0.24	0.47
M2: Cosine similarity of emoji to word difference ($S_2 = Cos(V_{Emoji}, V_{w_E} - V_{w_{\neg E}})$) vs. E								
corr(S_2, E)	Anger	Fear	Anticipation	Surprise	Joy	Sadness	Trust	Disgust
TurkishBERTweet	0.42	−0.36	−0.20	0.27	0.44	0.55	0.43	0.60
BERTurk	−0.07	0.14	−0.19	0.03	0.47	0.56	0.31	0.66
M3: Correlation between human annotated emotion score E and its opposite emotion $\neg E$								
corr(E, $\neg E$)	Anger vs. Fear		Surprise vs. Anticipation		Joy vs. Sadness		Disgust vs. Trust	
Annotation	0.76		0.46		−0.69		−0.59	

example, the *anger–fear* and *surprise–anticipation* pairs show positive correlations. However, the *joy–sadness* (−0.69) and *disgust–trust* (−0.59) pairs indicate moderate correlations in the expected direction.

Considering the associations between emotions, we conducted another measurement in M2, where we project each emoji vector along the spectrum of emotion pairs. We take vectors representing these emotion words as pairs (V_{w_E} and $V_{w_{\neg E}}$) and the cosine similarity between an emoji and the difference vector ($V_{w_E} - V_{w_{\neg E}}$) helps locate emojis along the emotion spectrum from E to $\neg E$. In this setting, TurkishBERTweet performs better for *anger*, *surprise*, and *trust*, and achieves comparable performance for *joy* and *sadness*.

5.1.2. Detecting cryptocurrencies from cashtags

Another case study involving the TurkishBERTweet preprocessor focuses on the use of cashtags. Cashtags are special entities on social media platforms, marked with a $ sign, that users employ to indicate specific fiat currencies and, more recently, cryptocurrencies (Cresci, Lillo, Regoli, Tardelli, & Tesconi, 2019; Hentschel & Alonso, 2014). In this study, we aim to explore the representations learned by TurkishBERTweet after preprocessing, as well as the standard vector embeddings obtained from the BERTurk model.

We collected unicode symbols or short codes for 30 fiat currencies[17] of different countries such as Dollar and Euro, as well as 30 Cryptocurrencies[18] like Bitcoin and Etherium based on their market cap size. We build three sets of entities: cryptocurrency symbols, fiat currency codes and their corresponding symbols.

Using the representations learned for these symbols and codes, we analyzed the vector similarities and variability of their embedding vectors. A model with better representations for these entities should be able to distinguish them effectively. In Fig. 5(a,b), we present the PCA embeddings of the vectors. The BERTurk model tends to collapse fiat currency symbols and produces mixed representations for cryptocurrencies and fiat currency codes. In contrast, TurkishBERTweet better differentiates among these three categories. The PCA embeddings for these models capture 27.8% and 24.4% of the variability for TurkishBERTweet and BERTurk, respectively.

To quantify how well the models distinguish set of currencies, we conducted an additional experiment. We selected a set of 10,000 random words as a baseline. We calculated pairwise cosine similarities within the sets and between different sets such as embeddings of cryptocurrencies and random words. We expect similarities within the same sets should be higher compared to similarities between different sets. In Fig. 5(c,d), we show distributions of cosine similarities for different set comparisons. TurkishBERTweet model achieves higher self-similarity for crypto and fiat symbols, while presenting the desired variability among each other. The distributions also significantly differ than similarities between random vectors. However, similarity distributions for BERTurk model are very similar to pairwise similarities of random words, indicating that the model do not capture meaningful representations for these entities.

5.2. Applications of LLM

One of the main use cases of the Language model, which we have presented in this article, is its use in research projects dealing with large amounts of data. Since TurkishBERTweet is an open source model with better performance and faster inference, it is a good choice for social media analysis projects. Here we can present an example of analyzing the sentiments of the dataset #Secim2023 (Najafi et al., 2024), which contains over 336 million tweets ranging from July 2021 to June 2023. Fig. 6 presents the daily aggregated sentiment deviances, and the daily aggregated sentiments for a year. The dates with extreme sentiment values are also mentioned in the figure. For instance, Feb 6, 2023, is highly negative, as a result of an unfortunate Turkey–Syria earthquake happened south-east part of the Türkiye.

5.3. Cost estimation

Based on OpenAI's pricing policy as of May 2024,[19] cost of inference using GPT3.5 Turbo model differs for input and output tokens. For one million tokens, the inputs and outputs cost C_{Input} = \$0.5 and C_{Output} = \$1.5, respectively. These amounts may change in the future since there are more companies offering similar services, devices getting more efficient, and OpenAI may change their marketing strategy.

The Eq. (1) consists of two parts: the input cost ($InferenceCost_{Input}$) and the output cost ($InferenceCost_{Output}$) per tweet. To perform a task using GPT3.5 Turbo model, we provide content from a tweet that has N_{tokens} tokens as an input. The model will return the classification outcome as one of the labels defined in the task encapsulated with BOS and EOS tokens, which results with three tokens per output.

$$\begin{aligned} &InferenceCost_{Input} = N_{tokens} * C_{Input} * 10^{-6} \\ &InferenceCost_{Output} = 3 * C_{Output} * 10^{-6} \\ &TotalInferenceCost = InferenceCost_{Input} + InferenceCost_{Output} \end{aligned} \tag{1}$$

For the dataset mentioned in the previous section, the number of tokens in the Election dataset using OpenAI's token counter is over 40.2 billion for more than 336 million tweets, which means that only the inference tasks cost nearly \$21K at the time of this publication. Considering the extreme budget requirement of commercial models, free alternatives such as TurkishBERTweet model offers the same

[17] https://fiatmarketcap.com/.

[18] https://coinmarketcap.com/.

[19] https://openai.com/pricing.

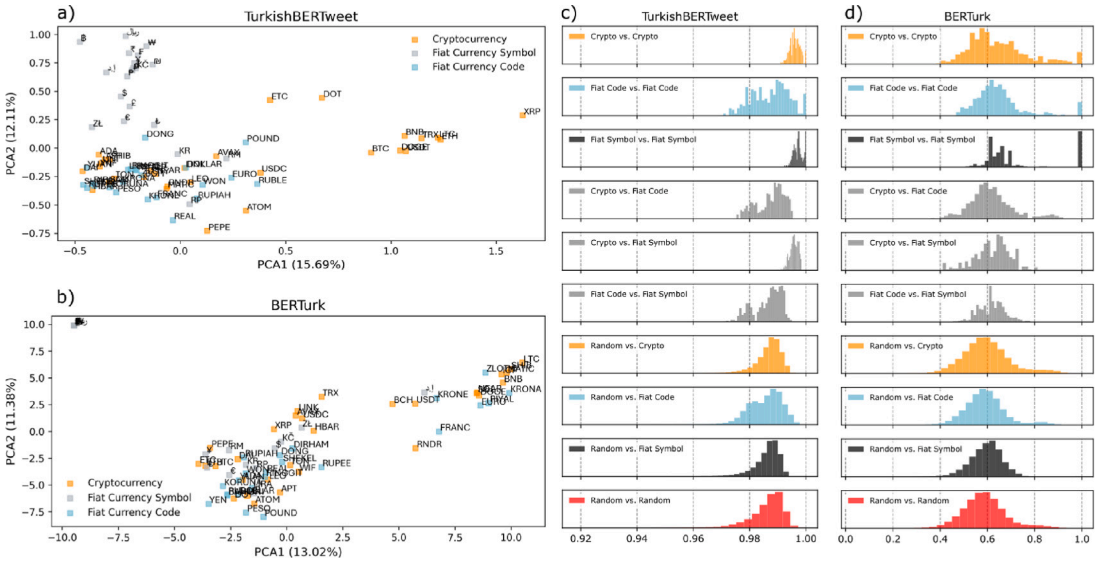


Fig. 5. Representation of fiat and cryptocurrencies. Learned embedding vectors for different currencies presented as 2-dimensional PCA embeddings (a,b). Vector similarities compared within and between the same groups, as well as random word vector embeddings to investigate quality of representations (c,d).

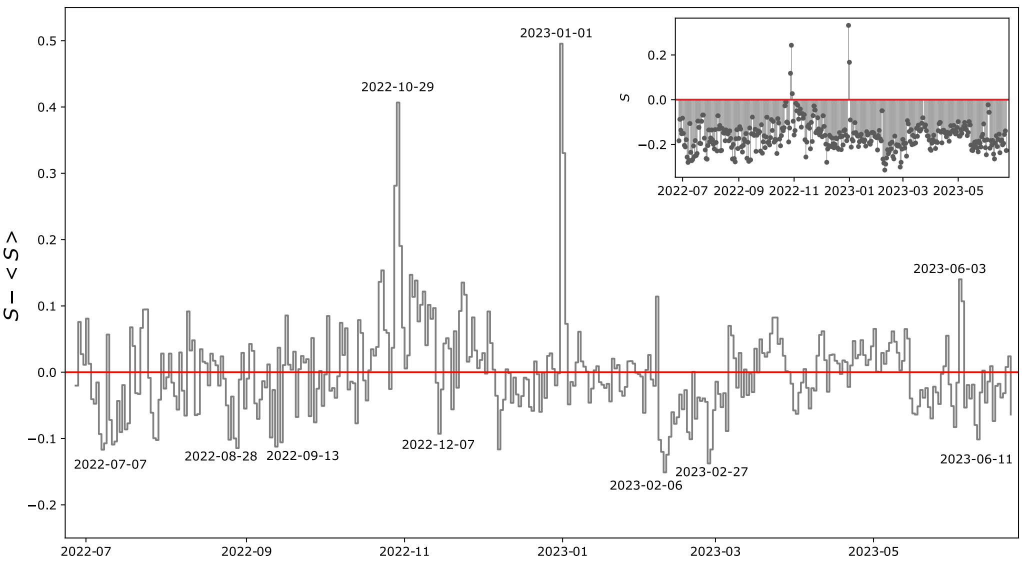


Fig. 6. Daily sentiment, and sentiment difference from the mean. We calculated sentiment timeseries for a year. S and $<S>$ stand for sentiment and mean sentiment of tweets and important social events observed in this period labeled.

performance. For our leave-one-dataset-out experiments reported in Table 4, the fine-tuning of those models cost more than 240$. The latest ChatGPT-4o model charge customers at much higher rate currently for million tokens (C_{Input} = \$5 and C_{Output} = \$15). Other available models like Google's Gemini 1.0 Pro offers different tiers for API usage. Free version offers significantly lower rate-limits and user-provided data can be used in training. More suitable paid option currently charges for prompts shorter than 128k tokens $C_{Input} = \$0.0875$ and $C_{Output} = \$1.05$.[20]

5.4. Limitations and future work

Turkish is one of the most widely used languages on social media platforms. There are problems require models and datasets to address those challenges. For instance, there are no open source Twitter datasets for tasks such as Named Entity Recognition and Part-of-speech tagging. There are only few papers on the task and they only share Tweet IDs (Küçük & Can, 2019), which prevented us from further comparisons with the baseline models. We want to evaluate performance of TurkishBERTweet on tasks other than task classification.

Embedding of the TurkishBERTweet can be used to classify social media posts by the emotions conveyed. Since our tokenizer can treat emojis specially, performance of emotion detection task can positively influenced by it. Another important task to study political tweet is to detect political tweets and the ideologies of users. One can use TurkishBERTweet to train models for these tasks.

Especially after Elon Musk's acquisition of Twitter/X, researchers are studying other platforms like TikTok and Instagram. We leave cross-platform comparisons as a future work, but TurkishBERTweet performance on generalizability task shows promise in that direction. To achieve that we are also planning to incorporate standard text and spend more effort to clean social media messages used in training stage.

Lastly, we can see a potential use case of TurkishBERTweet for detecting AI-generated content. Especially in social media, social bots can

[20] https://ai.google.dev/pricing.

utilize LLMs for creating content to manipulate discourse or interacting with real accounts (Yang et al., 2019).

6. Conclusions

TurkishBERTweet is the first language model pre-trained on over 894 million Turkish tweets. We introduced this language model for the Turkish NLP community, since it provides significant performance and suitable for large-scale analysis. The extensive experiments consider two different text classification tasks on 8 different datasets.

This work offers one of the most comprehensive benchmarks for Turkish NLP. We present results and comparisons for a diverse set of models. The rapidly evolving nature of the field makes it challenging to present up-to-date results, especially with the recent introduction of new LLM models. While industry and academic research groups continue to develop larger and better-performing models that can perform on multiple tasks at the same time, our findings show that none have surpassed the performance of models fine-tuned for specific tasks, yet. Moreover, their longer inference times and higher costs make them less preferable for large-scale analysis. Research community also face challenges to use some of the publicly available models since they may require resources beyond standard consumer-level GPUs available for researchers.

The novel experiments conducted by testing models on separate datasets shows generalizability of the TurkishBERTweet model. Also, TurkishBERTweet is a lightweight model that is computationally very efficient, so researchers can easily use it for their research tasks. Moreover, we showed that for data-extensive research that needs a significant amount of inferences, API-based models are costly. As they are close source, we also required to share our data with these platforms to be able to use them, which is a downside, especially when dealing with sensitive data.

7. Reproducibility

We hope that our publicly shared models will support research activities and adoption of it will lead to significant outcomes for social media research. Our `TurkishBERTweet` pre-trained models and LoRA adaptors are accessible on HuggingFace and code for preprocessor is available on Github. We are also providing the scripts, configurations used in the experimental sections and the results obtained for each model. Others can use these scripts to fine-tune and experiment with the wide collection of models presented in this work. All material offered for reproducibility can be accessed below:

- HuggingFace models: huggingface.co/VRLLab/TurkishBERTweet
- Preprocessor: github.com/ViralLab/TurkishBERTweet
- Experiment: github.com/ViralLab/TurkishBERTweetExperiments

CRediT authorship contribution statement

Ali Najafi: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Visualization. **Onur Varol:** Conceptualization, Methodology, Formal analysis, Investigation, Writing – review & editing, Visualization, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Onur Varol reports financial support was provided by TUBITAK National Observatory. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors are unable or have chosen not to specify which data has been used.

Acknowledgments

We thank Fatih Amasyali for providing access to Tweet Sentiment datasets from Kemik group. This material is based upon work supported by the Google Cloud Research Credits program with the award GCP19980904. We also thank TUBITAK (121C220 and 222N311) for funding this project.

References

Abadji, J., Ortiz Suarez, P., Romary, L., & Sagot, B. (2022). Towards a cleaner document-oriented multilingual crawled corpus. arXiv e-prints arXiv:2201.06642.

Abadji, J., Suárez, P. J. O., Romary, L., & Sagot, B. (2021). In H. Lüngen, M. Kupietz, P. Bański, A. Barbaresi, S. Clematide, & I. Pisetta (Eds.), *Proceedings of the workshop on challenges in the management of large corpora (CMLC-9) 2021. limerick, 12 July 2021 (online-event)*, *Ungoliant: An optimized pipeline for the generation of a very large-scale multilingual web corpus* (pp. 1–9). Mannheim: Leibniz-Institut für Deutsche Sprache, http://dx.doi.org/10.14618/ids-pub-10468, URL https://nbn-resolving.org/urn:nbn:de:bsz:mh39-104688.

AI@Meta (2024). Llama 3 model card. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Alecakir, H., Bölücü, N., & Can, B. (2022). *TurkishDelightNLP: A neural Turkish NLP toolkit*. ACL.

Alshaabi, T., Dewhurst, D. R., Minot, J. R., Arnold, M. V., Adams, J. L., Danforth, C. M., et al. (2021). The growing amplification of social media: Measuring temporal and social contagion dynamics for over 150 languages on Twitter for 2009–2020. *EPJ Data Science*, *10*(1), 15.

Amasyali, M. F., Tasköprü, H., & Çaliskan, K. (2018). Words, meanings, characters in sentiment analysis. In *2018 innovations in intelligent systems and applications conference* (pp. 1–6). IEEE.

Arın, İ., Işık, Z., Kutal, S., Dehghan, S., Özgür, A., & Yanikoğlu, B. (2023). SIU2023-NST-hate speech detection contest. In *2023 31st signal processing and communications applications conference* (pp. 1–4). IEEE.

Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., et al. (2023). Qwen technical report. arXiv preprint arXiv:2309.16609.

Baldwin, T., Cook, P., Lui, M., MacKinlay, A., & Wang, L. (2013). How noisy social media text, how diffrnt social media sources? In *Proceedings of the sixth international joint conference on natural language processing* (pp. 356–364).

Bas, O., Ogan, C. L., & Varol, O. (2022). The role of legacy media and social media in increasing public engagement about violence against women in Turkey. *Social Media+ Society*, *8*(4), Article 20563051221138939.

Çarık, B., & Yeniterzi, R. (2022). A Twitter corpus for named entity recognition in Turkish. In *Proceedings of the thirteenth language resources and evaluation conference* (pp. 4546–4551).

Caswell, I., Kreutzer, J., Wang, L., Wahab, A., van Esch, D., Ulzii-Orshikh, N., et al. (2021). Quality at a glance: An audit of web-crawled multilingual datasets. arXiv e-prints arXiv:2103.12028.

Çetin, M., & Amasyalı, M. F. (2013). Supervised and traditional term weighting methods for sentiment analysis. In *2013 21st signal processing and communications applications conference* (pp. 1–4). IEEE.

Cresci, S., Lillo, F., Regoli, D., Tardelli, S., & Tesconi, M. (2019). Cashtag piggybacking: Uncovering spam and bot activity in stock microblogs on Twitter. *ACM Transactions on the Web (TWEB)*, *13*(2), 1–27.

Derks, D., Fischer, A. H., & Bos, A. E. (2008). The role of emotion in computer-mediated communication: A review. *Computers in Human Behavior*, *24*(3), 766–785.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805 . arXiv:1810.04805, URL http://arxiv.org/abs/1810.04805.

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). Minneapolis, Minnesota: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N19-1423, URL https://aclanthology.org/N19-1423.

Farzindar, A., Inkpen, D., & Hirst, G. (2015). *Natural language processing for social media*. Springer.

Harlow, S. (2012). Social media and social movements: Facebook and an online guatemalan justice movement that moved offline. *New media & society*, *14*(2), 225–243.

Hentschel, M., & Alonso, O. (2014). Follow the money: A study of cashtags on Twitter. *First Monday*.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., et al. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., et al. (2023). Mistral 7B. arXiv preprint arXiv:2310.06825.

Jiang, Z.-H., Yu, W., Zhou, D., Chen, Y., Feng, J., & Yan, S. (2020). Convbert: Improving bert with span-based dynamic convolution. *Advances in Neural Information Processing Systems*, *33*, 12837–12848.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Köksal, A., & Özgür, A. (2021). Twitter dataset and evaluation of transformers for Turkish sentiment analysis. In *2021 29th signal processing and communications applications conference*.

Kralj Novak, P., Smailović, J., Sluban, B., & Mozetič, I. (2015). Sentiment of emojis. *PLoS One*, *10*(12), Article e0144296.

Küçük, D., & Can, F. (2019). A tweet dataset annotated for named entity recognition and stance detection. arXiv preprint arXiv:1901.04787.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., et al. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Longpre, S., Yauney, G., Reif, E., Lee, K., Roberts, A., Zoph, B., et al. (2023). A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. arXiv preprint arXiv:2305.13169.

Mangrulkar, S., Gugger, S., Debut, L., Belkada, Y., & Paul, S. (2022). PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Najafi, A., Mugurtay, N., Zouzou, Y., Demirci, E., Demirkiran, S., Karadeniz, H. A., et al. (2024). First public dataset to study 2023 Turkish general election. *Scientific Reports*, *14*(8794), http://dx.doi.org/10.1038/s41598-024-58006-w, URL https://doi.org/10.1038/s41598-024-58006-w.

Najafi, A., & Varol, O. (2024). Vrllab at HSD-2lang 2024: Turkish hate speech detection online with TurkishBERTweet. In *Proceedings of the 7th workshop on challenges and applications of automated extraction of socio-political events from text* CASE 2024, (pp. 185–189).

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

Ogan, C., & Varol, O. (2017). What is gained and what is left to be done when content analysis is added to network analysis in the study of a social movement: Twitter use during gezi park. *Information, Communication & Society*, *20*(8), 1220–1238.

OpenAI (2023). ChatGPT. URL https://chat.openai.com/.

Ortiz Su'arez, P. J., Romary, L., & Sagot, B. (2020). A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 1703–1714). Online: Association for Computational Linguistics, URL https://www.aclweb.org/anthology/2020.acl-main.156.

Ortiz Su'arez, P. J., Sagot, B., & Romary, L. (2019). In P. Bański, A. Barbaresi, H. Biber, E. Breiteneder, S. Clematide, M. Kupietz, H. L"ungen, & C. Iliadi (Eds.), *Proceedings of the workshop on challenges in the management of large corpora (CMLC-7) 2019. cardiff, 22nd July 2019*, *Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures* (pp. 9–16). Mannheim: Leibniz-Institut f"ur Deutsche Sprache, http://dx.doi.org/10.14618/ids-pub-9021, URL http://nbn-resolving.de/urn:nbn:de:bsz:mh39-90215.

Pfeffer, J., Matter, D., Jaidka, K., Varol, O., Mashhadi, A., Lasser, J., et al. (2023). Just another day on Twitter: a complete 24 hours of Twitter data. *Vol. 17*, In *Proceedings of the international AAAI conference on web and social media* (pp. 1073–1081).

Plutchik, R., & Kellerman, H. (1980). *Emotion, theory, research, and experience: theory, research and experience*. Academic Press.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., & Finn, C. (2024). Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, *36*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*(1), 5485–5551.

Schweter, S. (2020). BERTurk - BERT models for Turkish. http://dx.doi.org/10.5281/zenodo.3770924, URL https://doi.org/10.5281/zenodo.3770924.

Seckin, O. C., Atalay, A., Otenen, E., Duygu, U., & Varol, O. (2024). Mechanisms driving online vaccine debate during the COVID-19 pandemic. *Social Media+ Society*, *10*(1), Article 20563051241229657.

Segerberg, A., & Bennett, W. L. (2011). Social media and the organization of collective action: Using Twitter to explore the ecologies of two climate change protests. *The Communication Review*, *14*(3), 197–215.

Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909.

Sennrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 1715–1725). Berlin, Germany: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P16-1162, URL https://aclanthology.org/P16-1162.

Shoeb, A. A. M., & de Melo, G. (2020). EmoTag1200: Understanding the association between emojis and emotions. In *Proceedings of the 2020 conference on empirical methods in natural language processing* (pp. 8957–8967). Online: Association for Computational Linguistics, URL https://www.aclweb.org/anthology/2020.emnlp-main.720.

Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.-B., Yu, J., et al. (2023). Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805.

Toprak Kesgin, H., Yuce, M. K., & Amasyali, M. F. (2023). Developing and evaluating tiny to medium-sized Turkish BERT models. arXiv e-prints, arXiv–2307.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.

Uludoğan, G., Balal, Z. Y., Akkurt, F., Türker, M., Güngör, O., & Üsküdarlı, S. (2024). TURNA: A Turkish encoder-decoder language model for enhanced understanding and generation. arXiv preprint arXiv:2401.14373.

Uludoğan, G., Dehghan, S., Arın, I., Erol, E., Yanikoglu, B., & Özgür, A. (2024). Overview of the hate speech detection in Turkish and Arabic tweets (hsd-2lang) shared task at case 2024. In *Proceedings of the 7th workshop on challenges and applications of automated extraction of socio-political events from text* CASE 2024, (pp. 229–233).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., et al. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations* (pp. 38–45). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.emnlp-demos.6, URL https://aclanthology.org/2020.emnlp-demos.6.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., et al. (2020). mT5: A massively multilingual pre-trained text-to-text transformer. arXiv preprint arXiv:2010.11934.

Yang, K.-C., Varol, O., Davis, C. A., Ferrara, E., Flammini, A., & Menczer, F. (2019). Arming the public with artificial intelligence to counter social bots. *Human Behavior and Emerging Technologies*, *1*(1), 48–61.

Yang Liu, S., Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C. F., Cheng, K.-T., et al. (2024). Dora: Weight-decomposed low-rank adaptation. ArXiv, 5.