TURNA: A Turkish Encoder-Decoder Language Model for Enhanced Understanding and Generation

Gökçe Uludoğan[1] and **Zeynep Yirmibeşoğlu Balal**[1] and **Furkan Akkurt**[1]
Melikşah Türker[1,2] and **Onur Güngör**[1] and **Susan Üsküdarlı**[1]
[1]Department of Computer Engineering, Bogazici University
[2] VNGRS-AI
{gokce.uludogan,furkan.akkurt,zeynep.yirmibesoglu}@bogazici.edu.tr,
meliksah.turker@std.bogazici.edu.tr,onurgu@pt.bogazici.edu.tr,
suzan.uskudarli@bogazici.edu.tr

Abstract

The recent advances in natural language processing have predominantly favored well-resourced English-centric models, resulting in a significant gap with low-resource languages. In this work, we introduce TURNA, a language model developed for the low-resource language Turkish and is capable of both natural language understanding and generation tasks. TURNA is pretrained with an encoder-decoder architecture based on the unified framework UL2 with a diverse corpus that we specifically curated for this purpose. We evaluated TURNA with three generation and five understanding tasks for Turkish. The results show that TURNA outperforms several multilingual models in both understanding and generation tasks, and competes with monolingual Turkish models in understanding tasks.

1 Introduction

Recent advances in natural language processing (NLP) have predominantly resulted in English-centric models (Devlin et al., 2019; Clark et al., 2020; Radford et al., 2019; Brown et al., 2020; Touvron et al., 2023; Jiang et al., 2023), which have benefited from the vast amount of training data gathered from an abundance of English resources present on the web. The use of these models fuels further research yielding state-of-the-art results across various tasks (Touvron et al., 2023; Jiang et al., 2023). On the other hand, low-resource languages suffer from lack of data and limited computational resources, leading to a significant gap between models trained on well-resourced versus low-resource languages. Several multilingual models have been proposed to bridge this gap (Devlin et al., 2019; Conneau et al., 2020; Xue et al., 2021a; Liu et al., 2020). While such models address some tasks, they often fall short in those requiring deep understanding of language-specific nuances, such as dependency parsing and named entity recognition (Virtanen et al., 2019; Baumann, 2019; Tanvir et al., 2021). Thus, multilingual models lag behind monolingual models of the same scale (Rust et al., 2021; Nozza et al., 2020).

Recently, pretrained language models built upon transformers (Vaswani et al., 2017) have dominated NLP. These models vary in terms of their architectures and objectives. The architectures are commonly classified as encoder-only, decoder-only, or encoder-decoder models. Encoder-only models are typically trained with denoising objectives and focus on natural language understanding (NLU) tasks (Devlin et al., 2019; Clark et al., 2020). Decoder-only models are designed for natural language generation (NLG) tasks with causal language modeling (Radford et al., 2019; Brown et al., 2020; Touvron et al., 2023). Finally, encoder-decoder models deal with NLP tasks that require both NLU and NLG (Dong et al., 2019; Tay et al., 2023). Towards this end, the Text-to-Text Transformer (T5) (Raffel et al., 2020) employs an encoder-decoder architecture that is pretrained with a denoising objective known as span corruption. The Unifying Language Learning (UL2) framework (Tay et al., 2023) proposes the Mixture-of-Denoisers (MoD) pretraining objective which combines several denoising objectives. By coupling the MoD objective with an encoder-decoder architecture, state-of-the-art results are achieved in a range of NLP tasks.

For the Turkish language, low-resource in pretrained language models, encoder-only models exist (Schweter, 2020), however, there is a need for large-scale pretrained models that can perform both NLU and NLG. This work aims to develop such a model for Turkish that performs well across a variety of tasks of both types. Towards this end, we first compile a diverse Turkish corpus for pretraining purposes that includes web data, scientific articles, graduate theses, books, creative writing, and parliamentary speech transcriptions. Subsequently, we pretrain TURNA on this corpus with

Findings of the Association for Computational Linguistics: ACL 2024, pages 10103–10117
August 11-16, 2024 ©2024 Association for Computational Linguistics

an encoder-decoder architecture based on the UL2 framework (Tay et al., 2023). We report performance results for various generation and understanding tasks for Turkish. Our contributions are summarized as follows:

- The release of TURNA[1], the first unified language model capable of both understanding and generation tasks in Turkish. Thus far, this model is the largest of its kind, which has 1.1B parameters and is trained on a diverse range of corpora consisting of ∼43B tokens from various domains.
- The evaluation of TURNA on 13 datasets across eight tasks where it surpasses multilingual models across many tasks and it either outperforms or is on par with the state-of-the-art Turkish monolingual encoder-only model, BERTurk (Schweter, 2020)), in understanding tasks.
- The release of open-source code for collecting and filtering data[2], pretraining a monolingual model for Turkish[3], and fine-tuning this model for various tasks[4]. All resources are carefully prepared for the benefit of the scientific community with hopes of the furtherance of this work, model training and fine-tuning.
- A public and easy-to-use deployment of the model for all tasks presented in this paper[5].

2 Related Work

Multilingual language models address multiple languages including those considered low-resource languages. Turkish, considered as a low-resource language in this respect, is moderately represented in multilingual models such as mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020), mBART (Liu et al., 2020), mT5 (Xue et al., 2021a), XGLM (Lin et al., 2022), mGPT (Shliazhko et al., 2022), and mDeBERTa (He et al., 2023). However, these models are not up to par in language-specific tasks when compared with monolingual models developed with abundant data (Rust et al., 2021; Nozza et al., 2020).

A series of BERT models for Turkish known as BERTurk have been trained (Schweter, 2020) including several variations of BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2019), ConvBERTurk (Jiang et al., 2020), and ELECTRA (Clark et al., 2020). Most of these models were trained on a 35GB corpus consisting of 4.4B tokens drawn from the Turkish OSCAR corpus (Abadji et al., 2022), a Wikipedia dump[6], and various OPUS corpora (Tiedemann, 2012). Some models, like ConvBERTurk and ELECTRA, were also trained on the Turkish portion of the mC4 (Xue et al., 2021b) corpus – a certain cleaned version of the public web crawl data of Common Crawl. These models have been evaluated on various downstream tasks (such as part-of-speech tagging, named entity recognition, and question answering) where they generally outperform their multilingual counterparts mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020). These models are encoder-only, meaning mostly suitable for NLU tasks. Presently, there is a clear need for Turkish models that also excel in NLG tasks, which require a decoder component. This work focuses on an encoder-decoder model to address both types of tasks.

Encoder-only models are typically trained with span corruption with various lengths and frequencies (Devlin et al., 2019; Clark et al., 2020). Conversely, decoder-only models typically employ causal language modeling (Radford et al., 2019; Brown et al., 2020; Touvron et al., 2023) and are designed for generation tasks.

A popular model built on the transformer architecture, the Text-to-Text Transformer (T5) (Raffel et al., 2020), proposed a unified framework that treats all NLP tasks as conversions from some text to another. It employs an encoder-decoder architecture that is pretrained with a denoising objective. This model has demonstrated success over numerous tasks and is reported to scale well. UniLM (Dong et al., 2019) is also an encoder-decoder model, but pretrained using unidirectional, bidirectional, and sequence-to-sequence language modeling. This can be seen as a combination of causal and denoising objectives. The Unifying Language Learning framework (UL2) that is

[1] At https://huggingface.co (kept anonymous during the review process)

[2] https://anonymous.4open.science/r/turkish-academic-text-harvest

[3] https://anonymous.4open.science/r/TURNA-6753

[4] https://anonymous.4open.science/r/turkish-lm-tuner-37AF

[5] At https://huggingface.co (kept anonymous during the review process)

[6] An unspecified Wikipedia dump which we speculate to be from 2020.

based on a pretraining objective called Mixture-of-Denoisers (MoD) was proposed, which combines span corruption objectives with varying mixture parameters (Tay et al., 2023). This study found that among the decoder-only and encoder-decoder models, both of which are trained using the MoD objectives, the encoder-decoder models performed better. By using the MoD objective and moderately scaling up the model, they achieved state-of-the-art performance on a diverse set of NLU and NLG tasks.

3 Data

We compiled a diverse Turkish monolingual dataset to pretrain our model. Our dataset comprises of a web corpus, scientific corpora consisting of Turkish articles and graduate theses, Turkish books, a corpus of creative writing assignments from Bilkent University, and transcriptions of parliamentary debates. The details of each corpus are explained in the following subsections, and the training corpora statistics are summarized in Table 1.

During data splitting, we ensured that the validation set of each dataset contained a minimum of 100K tokens. The train-validation splits are reported under each subsection.

3.1 Web Corpora

mC4 (Raffel et al., 2020) and OSCAR-2201 (Abadji et al., 2022) are two large multilingual web corpora. Their Turkish sections contain 87.7M and 10.8M web pages, yielding 98.5M web pages in total. As is common in web content, this data includes noise, such as titles and repeating SEO (search engine optimization) targeted keywords that are not considered natural language. Therefore, such corpora should be cleaned before being used for training. The OSCAR and mC4 corpora used in this work were cleaned by the VNGRS-AI team using a set of heuristics and rules, detailed in their work (Turker et al., 2024a). The cleaned version of the combined web corpus contains 50.3M pages.

3.2 Scientific Corpora

To create a corpus in the scientific domain characterized by its formal and informative language style, we collected articles and theses written in Turkish. We downloaded the articles from DergiPark[7], a major platform for Turkish academic journals. Our initial collection included 407,146 articles, all in PDF format and labeled as Turkish. These articles were sourced from 1,857 distinct journals, comprising a diverse range of topics. These articles form our *Dergipark* scientific corpus.

In addition to articles, we also collected scientific texts in the form of theses. These theses, products of higher education in Turkey, were accessed through Turkey's National Thesis Center[8]. From this repository, we downloaded 486,166 theses marked as Turkish, which compose our *YökTez* scientific corpus.

The collected documents were in PDF format. For text extraction, we utilized the Apache Tika parser[9]. We applied a rigorous cleaning and filtering strategy to remove undesired content like page numbers, equations, table entries, and similar unnecessary tokens introduced by the extraction process, as detailed in Section A.1.

We used 99.99 of the cleaned *Dergipark* documents for training and the rest for validation, to avoid over-inflation of the validation set due to the high number of documents. For *YökTez*, 99.999 of the documents were used for training. The final number of documents and the number of tokens after line and document-wise filtering of our scientific training corpora are listed in Table 1.

3.3 Book Corpus

The *Book Corpus* is a compilation of 5,080 Turkish fiction and non-fiction books. We cleaned the Book Corpus in a similar fashion to the previously mentioned procedures, albeit with a simpler heuristic. We first standardized the punctuation and removed invalid characters. The initial 100 lines of each book have been filtered out if they contain author, translator, or publishing information. We dropped any line in each book with all numerals, or that contained a URL or an e-mail. After the initial 70% of lines, we truncated the lines after a keyword indicating a bibliography, notes, or a list of works of the author or the publishing house. 99.97% of the books were used for training (5,078 books), and the remaining two books were used for validation.

3.4 Bilkent Creative Writings

The *Bilkent Creative Writings* corpus comprises 8,630 documents produced by Bilkent University students while taking creative writing courses in

[7] dergipark.org.tr

[8] tez.yok.gov.tr/UlusalTezMerkezi

[9] github.com/apache/tika

Corpus	Type	# Docs	# Tokens (B)
OSCAR & mC4	Web	50,336,214	25.33
Dergipark	Scientific	334,429	1.78
Yöktez	Scientific	475,817	15.24
Books	Literary	5,078	0.61
Bilkent Creative Writings	Creative Text	8,457	0.01
ParlaMintTR	Dialogue	1,333	0.07

Table 1: Training Corpora Statistics

Turkish[10]. This data was cleaned similarly to the Book Corpus by removing special keywords (such as the Turkish word for assignment) and truncating the content after the bibliographies. 8,457 of them were used for training and the rest was used for validation.

3.5 ParlaMintTR

The *ParlaMintTR* corpus is assembled from the CLARIN Flagship project[11] and consists of the Turkish portion of parliamentary debates in Europe (1,335 documents). No special cleaning or filtering was applied to this data. 1,333 of the debates were used for training, and two were used for validation.

4 Methodology

4.1 Model

We used an encoder-decoder Transformer model[12] (Raffel et al., 2020) for TURNA. This choice was based on the finding that encoder-decoder models surpass decoder-only models when the UL2 objective is used, as demonstrated in Tay et al., 2023. Furthermore, the encoder component can still be employed effectively for understanding tasks when coupled with task-specific classification heads, thus reducing the model parameters by half. Due to our limited computational resources, we opted for the Large36L configuration (Tay et al., 2021) for our model. This configuration requires only 37% of the parameters of a model configuration of comparable size, yet still outperforms it.

TURNA has 36 encoder and decoder layers, each with 16 attention heads. The model's token embeddings are 1,024 dimensional. The multi-layer perceptron layers have 2,816 hidden dimensions and employ Gated GeLu activations (Shazeer, 2020). The parameters of the input and classification layers are not shared. These architectural choices result in a model with 1.1B parameters.

For tokenization, we used a unigram subword tokenizer (Kudo, 2018) trained on 10GB of text that consists of random subsets of OSCAR (Abadji et al., 2022), OPUS (Zhang et al., 2020) and a Wikipedia dump dated September 17, 2021, using the SentencePiece implementation[13] (Kudo and Richardson, 2018). This tokenizer[14] (Turker et al., 2024b) is provided by the VNGRS-AI Team. The initial vocabulary size of 32,000 was expanded to 32,128 with the addition of 128 sentinel tokens used by pretraining objectives.

4.2 Pretraining Objectives

The pretraining was performed with Mixture-of-Denoisers (MoD), consisting of several denoising objectives, which were shown to achieve better downstream performance (Tay et al., 2023). These objectives are R-denoising (regular denoising), S-denoising (sequential denoising), and X-denoising (extreme denoising), each characterized by the mean length of the corrupted spans, the ratio of corrupted tokens, and the number of corrupted spans. R-denoising follows the standard span corruption method of T5, selecting spans of 2 to 5 tokens, covering about 15% of the input. The task is then to predict the corrupted tokens in the decoder output. S-denoising, on the other hand, corrupts a continuous portion from a random point in the input, accounting for approximately 25% of the input. Similar to R-denoising, this objective aims to predict a single corrupted span. However, it is similar to standard causal language modeling in its modeling approach. X-denoising is designed as an interpolation between R-denoising and S-denoising. It aims to corrupt 50% of the input on average. This is achieved through a varying mix of many short or fewer long corrupted spans, exposing the model to both denoising and causal language modeling-like objectives. During pretraining, these objectives are randomly assigned to each input sequence, with a distribution of 40% each for the R- and X-denoisers and 20% for the S-denoiser.

The model differentiates between these denoisers by using specific sentinel tokens at the beginning of samples: `[NLG]` for the X-denoiser, `[NLU]` for the R-denoiser, and `[S2S]` for the S-denoiser.

[10] github.com/selimfirat/bilkent-turkish-writings-dataset

[11] clarin.eu/parlamint

[12] Specifically, we used the version 1.1 of the official T5 implementation described at github.com/google-research/text-to-text-transfer-transformer/blob/main/released_checkpoints.md#t511

[13] github.com/google/sentencepiece

[14] github.com/vngrs-ai/vnlp/tree/main/vnlp/turkish_word_embeddings

4.3 Implementation details

Pretraining. We pretrained TURNA for a total of 1,740,000 steps with a batch size of 48 and a source and target sequence length of 512 using a single v3-8 type TPU with the T5X[15] library. This configuration results in TURNA being exposed to 42.7B tokens at the end of its training. We disabled dropout during pretraining but enabled it during fine-tuning.

The pretraining data is a mixture of samples from the collected datasets. To ensure a fair representation of different language characteristics, we randomly selected samples from each dataset according to their proportions: Web Corpora (50%), YökTez (25%), DergiPark (10%), Book Corpus (10%), ParlaMintTR (3%), and Bilkent Creative Writings (2%).

Baselines. We compared our model with multilingual models: mT5, specifically mT5-large[16] (Xue et al., 2021a), and mBART[17] (Liu et al., 2020), as well as a monolingual encoder-only model, BERTurk[18] (Schweter, 2020), where applicable.

Fine-tuning. We fine-tuned the models using Hugging Face's `transformers` library[19] (Wolf et al., 2020) on NVIDIA A40 GPUs. The standard text-to-text formulation is used for fine-tuning the encoder-decoder models, i.e., TURNA, mT5 and mBART. Additionally, we fine-tuned TURNA's encoder with a task-specific head for certain understanding tasks, referring to it as TURNA-Encoder. The models were optimized for 10 epochs with an early stopping patience of 3 epochs. We used the Adafactor optimizer (Shazeer and Stern, 2018) with a learning rate of 1×10^{-3} to tune TURNA and mT5 models, without a scheduler. However, our attempts at fine-tuning the mBART model with the Adafactor optimizer did not yield a satisfactory training loss curve. Consequently, we opted for the AdamW optimizer (Loshchilov and Hutter, 2017) with a learning rate of 5×10^{-5} and a linear scheduler. The same optimizer and scheduler settings were applied for fine-tuning the BERTurk and TURNA-Encoder models. Due to our limited computational resources, we could not perform hyperparameter tuning and used the recommended fine-tuning settings for Adafactor[20] and default trainer settings[21] for AdamW. For each task and dataset, the batch size, and maximum input and target length parameters were individually selected, and their corresponding values can be found in Table 7.

We used beam decoding with a beam size of 4 and early stopping to generate predictions. For summarization and title generation tasks, we also applied a length penalty of 2 and enforced a no-repeat n-gram size of 3 to ensure the diversity of the output and prevent repetition of sequences.

5 Experiments

5.1 Fine-tuning tasks

This section provides an overview of downstream tasks used to evaluate our model. These tasks assess model capabilities across various domains, and include both natural language understanding and generation tasks. The understanding tasks include text classification, natural language inference, semantic textual similarity, named entity recognition, and part-of-speech tagging. The generation tasks comprise paraphrasing, summarization, and news title generation.

Paraphrasing. This task involves rephrasing a given text while retaining the original meaning. It assesses the model's understanding of semantics and its ability to generate diverse texts. We utilized two paraphrasing datasets, constructed from parallel corpora via machine translation and filtered based on semantic similarity (Alkurdi et al., 2022). They are TAT, which contains paraphrases from Tatoeba[22], and OST, which includes pairs from OpenSubtitles2018 (Lison et al., 2018).

Summarization. Similar to paraphrasing, summarization also rephrases a text. However, it aims to produce a condensed version that only includes key information. Consequently, it imposes additional constraints on a model's generative capabilities. For evaluation, we used two datasets: TRNews (Baykara and Güngör, 2022) and the Turkish subset of MLSUM (Scialom et al., 2020).

News Title Generation. Generating titles for news articles evaluates a model's ability to capture

[15] `github.com/google-research/t5x`
[16] `hf.co/google/mt5-large`
[17] `hf.co/facebook/mbart-large-cc25`
[18] `hf.co/dbmdz/bert-base-turkish-cased`
[19] `github.com/huggingface/transformers`
[20] `hf.co/docs/transformers/main_classes/optimizer_schedules#transformers.Adafactor`
[21] `hf.co/docs/transformers/main_classes/trainer#trainer`
[22] `tatoeba.org`

the most salient information in a concise manner and checks the model's creativity and understanding of key phrases in the news domain. We used the same two summarization datasets: TRNews and MLSUM.

Named Entity Recognition. Named entity recognition (NER) aims to locate named entities, and subsequently classifies these entities into predefined categories, typically "person", "location" and "organization". We employed two datasets for this task: WikiANN (Rahimi et al., 2019) and MilliyetNER (Tür et al., 2003).

Part-of-speech Tagging. Part-of-speech (POS) tagging involves categorizing each word in a sentence according to its grammatical function. This task assigns a specific part of speech, such as noun, pronoun, or verb, to each word, classifying its role within the structure of a sentence. We used two Turkish Universal Dependencies (Nivre et al., 2020) treebanks, IMST (Türk et al., 2023) and BOUN (Marşan et al., 2023), to fine-tune and evaluate our model.

Semantic Textual Similarity. Semantic textual similarity (STS) tests a model's ability to contextually compare two sentences by producing a similarity score. We used the STSb-TR (Beken Fikri et al., 2021) dataset to fine-tune and evaluate our model.

Natural Language Inference. Natural language inference (NLI), also known as textual entailment, involves examining a pair of sentences, the premise and the hypothesis, to determine their relationship as "entailment", "contradiction", or "neutral". This task tests a model's understanding of context by assessing if the hypothesis logically follows the premise. Therefore, NLI also measures a model's reasoning skills. For this task, we used the Natural Language Inference in Turkish (NLI-TR) dataset (Budur et al., 2020) for evaluation.

Text Classification. Text classification involves categorizing texts into predefined groups based on their contents. This task assesses the model's contextual awareness and robustness in extracting relevant features from the input text, allowing it to discern important patterns and information crucial for accurate classification. We used three different datasets for evaluating this task: Product Reviews[23], TTC4900[24] (Yıldırım and Yıldız, 2018), and Tweet Sentiments (Amasyali et al., 2018).

5.2 Evaluation Metrics

We evaluated the generation tasks with ROUGE (Lin, 2004), BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005) metrics. For the understanding tasks, we adopted standard classification metrics such as accuracy, precision, recall, and F1. The only exception was semantic textual similarity, a regression task, for which we used the Pearson correlation coefficient for evaluation. For NLI and classification tasks, weighted precision, recall and F1 were reported, leaving out accuracy due to its equality to weighted recall.

5.3 Results

5.3.1 Generation Tasks

We evaluated TURNA's generative capabilities on three tasks and compared the results to mT5 and mBART. The results, as presented in Table 2, show that TURNA outperformed the baseline models in both paraphrasing and summarization, with mT5 ranking second and mBART last. In title generation, TURNA performed the best on the TRNews dataset, followed by mBART. However, for the MLSUM dataset, mBART outperformed both TURNA and mT5.

Table 2: Downstream performance of models on generation tasks.

Task	Dataset	Model	Rouge1	Rouge2	RougeL	BLEU	METEOR
Paraphrasing	OST	mBART	76.86	61.34	75.18	48.85	72.61
		mT5	77.49	62.15	75.87	49.66	73.61
		TURNA	78.43	63.58	76.81	51.47	74.79
	TAT	mBART	82.77	68.68	81.31	55.57	77.34
		mT5	88.76	77.75	87.51	67.80	85.58
		TURNA	90.22	80.23	88.95	71.14	87.56
Summarization	MLSUM	mBART	41.39	27.63	35.61	19.66	32.30
		mT5	43.43	29.95	37.71	21.58	34.20
		TURNA	44.33	30.99	38.62	24.25	36.47
	TRNews	mBART	39.96	25.53	34.90	16.69	32.23
		mT5	41.46	27.47	36.60	18.31	34.48
		TURNA	41.77	27.81	36.99	19.05	34.61
Title Generation	MLSUM	mBART	32.97	19.71	31.32	7.41	18.29
		mT5	32.60	19.65	30.93	7.15	17.75
		TURNA	32.67	19.60	31.12	7.08	17.90
	TRNews	mBART	35.40	21.92	34.32	11.95	23.26
		mT5	34.84	21.62	33.85	11.96	22.40
		TURNA	36.47	22.88	35.47	12.64	23.62

5.3.2 Understanding Tasks

In assessing understanding tasks, we compared both encoder-decoder models fine-tuned with the

[23]hf.co/datasets/turkish_product_reviews
[24]kaggle.com/savasy/ttc4900

standard text-to-text formulation and encoder-only models, such as TURNA-Encoder and BERTurk. TURNA achieved results that surpass both mT5 and mBART across various tasks and datasets, as detailed in Tables 3, 4, and 5, reporting POS tagging & NER, NLI, and classification results, respectively. TURNA outperformed mBART and mT5 in all classification, NLI, STS, POS tagging and NER tasks, except for the Milliyet (NER) dataset. While TURNA slightly lagged behind BERTurk on some tasks, this was not surprising as encoder-decoder models often struggle with understanding tasks (Lewis et al., 2020; Kementchedjhieva and Chalkidis, 2023). However, TURNA-Encoder surpassed BERTurk in NER, NLI and some classification tasks, and was competitive in others. The notable exception was the semantic textual similarity task (Table 6), where TURNA-Encoder significantly lagged behind BERTurk. This suggests that further hyperparameter tuning could improve performance, as evidenced by an additional experiment where adjusting the learning rate enabled TURNA-Encoder to achieve a significantly higher Pearson correlation score in the STS task (refer to Table 11 in the Appendix).

Table 3: Downstream performance of models on POS tagging and NER.

Task	Dataset	Model	Precision	Recall	F1	Accuracy
POS	BOUN	mBART	88.15	87.75	87.95	87.75
		mT5	90.90	90.74	90.82	90.74
		TURNA	92.39	92.35	92.37	92.35
		BERTurk	90.60	90.41	90.50	93.22
		TURNA-Encoder	90.30	90.31	90.31	93.05
	IMST	mBART	77.68	77.40	77.54	77.39
		mT5	93.17	93.05	93.11	93.04
		TURNA	94.66	94.48	94.57	94.48
		BERTurk	94.28	94.14	94.21	95.62
		TURNA-Encoder	93.34	93.27	93.31	94.91
NER	Milliyet	mBART	87.62	70.67	78.23	98.11
		mT5	84.73	71.98	77.83	98.20
		TURNA	91.36	83.28	87.13	97.91
		BERTurk	93.51	94.84	94.17	99.24
		TURNA	95.16	96.03	95.59	99.46
	WikiANN	mBART	90.76	89.12	89.93	95.84
		mT5	90.50	89.90	90.20	95.93
		TURNA	90.48	90.20	90.34	96.18
		BERTurk	89.83	90.41	90.12	96.53
		TURNA-Encoder	91.08	92.01	91.54	97.08

Table 4: Downstream performance of models on natural language inference (NLI).

Model	Precision	Recall	F1
mBART	86.14	86.06	86.08
mT5	83.67	83.66	83.66
TURNA	86.20	86.19	86.19
BERTurk	86.94	86.88	86.90
TURNA-Encoder	88.28	88.30	88.28

Table 5: Downstream performance of models on text classification.

Dataset	Model	Precision	Recall	F1
Product Reviews	mBART	87.67	93.63	90.55
	mT5	93.01	94.17	93.27
	TURNA	94.67	95.24	94.81
	BERTurk	94.90	95.44	94.70
	TURNA-Encoder	95.57	95.92	95.67
TTC4900	mBART	78.23	71.81	73.08
	mT5	67.52	66.74	66.80
	TURNA	89.15	88.11	88.16
	BERTurk	91.97	91.85	91.88
	TURNA-Encoder	91.05	90.53	90.52
Tweet Sentiment	mBART	74.07	71.85	72.25
	mT5	68.20	67.45	66.71
	TURNA	74.58	73.78	73.94
	BERTurk	75.91	75.20	74.79
	TURNA-Encoder	77.08	76.82	76.76

Table 6: Downstream performance of models on semantic textual similarity (STS).

Model	Pearson
mBART	66.95
mT5	59.40
TURNA	78.74
BERTurk	82.60
TURNA-Encoder	73.63

6 Conclusion

In this study, we introduced TURNA, a new Turkish language model that adopts an encoder-decoder architecture following the UL2 framework. This model was pretrained on a broad corpus covering web data, scientific articles, theses, books, creative writing, and parliament corpora. Our comprehensive evaluations across three generation and five understanding tasks on 13 different datasets showed that TURNA outperforms existing multilingual models, mT5 and mBART, and performs better than or on par with the Turkish encoder-only model BERTurk. To encourage further research and facilitate benchmarking in Turkish NLP, these models and the entire source code for data collection, filtering, model training, and fine-tuning are made publicly accessible.

Limitations

TURNA, with its 1.1B parameters, excels in a variety of NLP tasks, surpassing similar-scale multilingual models like mT5 (1.2B) and mBART (610M) in both generation and understanding. However, its efficiency, especially in understanding tasks, is closely matched by the smaller, encoder-only model BERTurk, which has only 110M parameters. This suggests that the scale-to-performance ratio of TURNA may not be as efficient as expected.

Addressing this, we modified TURNA into TURNA-Encoder by removing the decoder and adding task-specific heads, which enhanced its efficiency. TURNA-Encoder, having half the parameters of TURNA, surpassed BERTurk in some tasks, showing an improvement in efficiency. However, the comparison with BERTurk indicates a need for additional pretraining to fully leverage TURNA's larger parameter count.

Current research on scaling laws indicates that training models for up to four epochs can be beneficial (Taylor et al., 2022; Muennighoff et al., 2023). Despite having 1.1B parameters, TURNA has been trained with approximately 43B tokens, which is roughly equivalent to one epoch. This undertraining might be limiting its potential. Therefore, we suggest further pretraining of TURNA to enhance its performance.

In our downstream evaluations, we used the same optimization hyperparameters across all tasks and datasets due to limited computational resources. This approach may have influenced performance as datasets carry differing sizes and tasks exhibit different difficulties. Hence, we suggest dataset and task-specific hyperparameter tuning to thoroughly demonstrate the capabilities of our model in downstream tasks.

Ethics Statement

Web content carries the risk of harmful content including toxicity, abuse, and obscenity. Significant effort was expended to remove such harmful language from the web corpus that we used to train TURNA. However, despite the efforts to filter out such content, there is a high risk that some of the harmful content still remains in the training corpora. Thus, such language could emerge during language generation tasks. This calls for continuous monitoring of this system to eliminate such occurrences.

Another concern is the introduction of bias into TURNA from the data we used for training. Such bias is significant when it concerns tasks that render decisions involving people, such as admission, promotion, and loans. More research is needed to detect and deal with biases such as based on gender, race, ethnicity, religion, and other social factors.

Incorporating books, theses, and papers into the training data concerns the licensing. We have released our model under a restricted license, permitting only academic use.

AI Assistants

During the coding of the model, we used GitHub Copilot[25] to write some of the boilerplate parts of the code, which is a timesaver when formulating standard constructs. Most of the code snippets were related to data processing scripts. The team members have written the code for all the core functionality of the data processing, the model, and the evaluations. All the code has gotten meticulously reviewed as part of handling pull requests.

ChatGPT[26] and Notion AI[27] services were utilized to proofread, spell-check, and correct the grammar of this document. These services were mostly utilized during the early stages of writing. The resulting manuscript has been carefully reviewed by team members for correctness, flow, and articulation.

References

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. 2022. Towards a Cleaner Document-Oriented Multilingual Crawled Corpus. *arXiv e-prints*, page arXiv:2201.06642.

Besher Alkurdi, Hasan Yunus Sarioglu, and Mehmet Fatih Amasyali. 2022. Semantic similarity based filtering for Turkish paraphrase dataset creation. In *Proceedings of the 5th International Conference on Natural Language and Speech Processing (ICNLSP 2022)*, pages 119–127, Trento, Italy. Association for Computational Linguistics.

Mehmet Fatih Amasyali, Hakan Tasköprü, and Kübra Çaliskan. 2018. Words, meanings, characters in sentiment analysis. In *2018 Innovations in Intelligent*

[25]https://github.com/features/copilot
[26]https://chat.openai.com/
[27]https://www.notion.so/product/ai

Systems and Applications Conference (ASYU), pages 1–6. IEEE.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Antonia Baumann. 2019. Multilingual language models for named entity recognition in German and English. In *Proceedings of the Student Research Workshop Associated with RANLP 2019*, pages 21–27, Varna, Bulgaria. INCOMA Ltd.

Batuhan Baykara and Tunga Güngör. 2022. Abstractive text summarization and new large-scale datasets for agglutinative languages turkish and hungarian. *Language Resources and Evaluation*, 56(3):973–1007.

Figen Beken Fikri, Kemal Oflazer, and Berrin Yanikoglu. 2021. Semantic similarity based evaluation for abstractive news summarization. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 24–33, Online. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Emrah Budur, Rıza Özçelik, Tunga Gungor, and Christopher Potts. 2020. Data and Representation for Turkish Natural Language Inference. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8253–8267, Online. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *Advances in neural information processing systems*, 32.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Zi-Hang Jiang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. 2020. Convbert: Improving bert with span-based dynamic convolution. *Advances in Neural Information Processing Systems*, 33:12837–12848.

Yova Kementchedjhieva and Ilias Chalkidis. 2023. An exploration of encoder-decoder approaches to multi-label classification for legal and biomedical text. In *Findings of the Association for Computational*

Linguistics: ACL 2023, pages 5828–5843, Toronto, Canada. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Büşra Marşan, Salih Furkan Akkurt, Utku Türk, Furkan Atmaca, Şaziye Betül Özateş, Gözde Berk, Seyyit Talha Bedir, Abdullatif Köksal, Balkız Öztürk Başaran, Tunga Güngör, and Arzucan Özgür. 2023. Ud turkish boun. Universal Dependencies. Accessed: 2023-11-14.

Niklas Muennighoff, Alexander M Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. 2023. Scaling data-constrained language models. *arXiv preprint arXiv:2305.16264*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2020. What the [mask]? making sense of language-specific bert models. *arXiv preprint arXiv:2003.02912*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. pages 311–318.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Stefan Schweter. 2020. Berturk - bert models for turkish.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. MLSUM: The multilingual summarization corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8051–8067, Online. Association for Computational Linguistics.

Noam Shazeer. 2020. Glu variants improve transformer.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mgpt: Few-shot learners go multilingual. *arXiv preprint arXiv:2204.07580*.

Umut Sulubacak and Gülşen Eryiğit. 2018. Implementing universal dependency, morphology, and multiword expression annotation standards for turkish language processing. *Turkish Journal of Electrical Engineering and Computer Sciences*, 26(3):1662–1672.

Hasan Tanvir, Claudia Kittask, Sandra Eiche, and Kairit Sirts. 2021. EstBERT: A pretrained language-specific BERT for Estonian. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 11–19, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. 2021. Scale efficiently: Insights from pre-training and fine-tuning transformers. *arXiv preprint arXiv:2109.10686*.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Neil Houlsby, and Donald Metzler. 2023. Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Meliksah Turker, Erdi Ari, and Aydin Han. 2024a. VBART: The Turkish LLM. *arXiv preprint arXiv:2403.01308*.

Meliksah Turker, Erdi Ari, and Aydin Han. 2024b. VNLP: Turkish NLP package. *arXiv preprint arXiv:2403.01309*.

Gökhan Tür, Dilek Hakkani-Tür, and Kemal Oflazer. 2003. A statistical information extraction system for turkish. *Natural Language Engineering*, 9(2):181–210.

Utku Türk, Şaziye Betül Özateş, Büşra Marşan, Salih Furkan Akkurt, Çağrı Çöltekin, Gülşen Cebiroğlu Eryiğit, Memduh Gökırmak, Hüner Kaşıkara, Umut Sulubacak, and Francis Tyers. 2023. Ud turkish imst. Universal Dependencies. Accessed: 2023-11-14.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: Bert for finnish. *arXiv preprint arXiv:1912.07076*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021a. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021b. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Savaş Yıldırım and Tuğba Yıldız. 2018. A comparative analysis of text classification for turkish language. *Pamukkale Univ Muh Bilim Derg*, 24(5):879–886. Doi: 10.5505/pajes.2018.15931.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation.

A Appendix

A.1 Cleaning Procedure for Scientific Corpus

Initially, we replaced invalid or misinterpreted characters resulting from Optical Character Recognition (OCR) errors, employing a predefined dictionary. Subsequently, we omitted preliminary text appearing before the abstract, which typically contains non-essential information such as affiliations and article metadata. This was achieved using regular expressions tailored for this purpose. While this approach was sufficient for scientific articles, the theses posed additional challenges, including sections like lists of figures, tables, and customary declarations. To handle these sections, we relied on regular expressions designed to identify and subsequently discard specific titles and their accompanying content.

In our effort to maintain the quality of the extracted text from the PDF articles, we also implemented a line-wise filtering procedure involving the steps below:

- **Text Statistics:** Each line from the articles was analyzed based on various statistics. These included character count, token count, numeric content, average token length, and metrics reflecting the prevalence of numbers, specifically the proportion of numeric tokens to total tokens and frequency of digit appearances. This stage ensured the removal of non-content elements, such as headers, page numbers, and table items.

- **Language Identification and Correction:** Given the potential presence of non-Turkish lines within the articles, each line was checked for its Turkish content using the `langid` library[28]. In cases of potential anomalies or false detections, the surrounding lines were examined to correct such anomalies, ensuring that the majority of our extracted content is in Turkish.

- **Content Identification:** Although article metadata typically appears at the beginning of the documents, they may also appear elsewhere. To identify such elements as dates, email addresses, and names, each line was checked using specific regular expressions. Additionally, captions, identified by their distinct patterns, were detected and subsequently removed.

- **Identification and Filtering of Special Sections:** In scientific texts, certain lines—like those in bibliographies and footnotes—may not contribute essential content, or they may even disrupt the primary narrative. To address this, we implemented strategies to detect and subsequently omit such lines. This step ensured the retention of the text's coherence and continuity.

- **Citation Filtering:** Citations, while crucial to academic papers, can interrupt text flow, especially when preparing data for language model training. We thus used patterns to identify and remove inline citations, guaranteeing a smooth textual flow.

After the line-wise filtering procedure was complete, we applied document-based filtering with the help of a Statistical Language Model (LM) trained on a compilation of May 2023 Turkish Wikipedia articles[29]. A KenLM 5-gram language model was trained (Heafield, 2011) on 6.8M sentences tokenized with a Turkish SentencePiece tokenizer[30]. The KenLM model was then used to discard documents defined by separate thresholds for the *Dergipark* articles (less than 5% LM score) and the *YökTez* theses (less than 2% LM score). The thresholds have been selected by native Turkish speakers by analyzing the distribution of documents and their qualities based on document-based average LM score.

A.2 Fine-tuning Datasets

OST (Alkurdi et al., 2022) OST is a paraphrasing dataset, constructed by translating English sub-

[28] `github.com/saffsd/langid.py`

[29] `hf.co/datasets/musabg/wikipedia-tr`

[30] `github.com/vngrs-ai/vnlp/tree/main/vnlp/turkish_word_embeddings`

titles from OpenSubtitles2018 (Lison et al., 2018) into Turkish. The original subtitles and their translations were preprocessed to create an unfiltered version of the dataset with 1,944,955 pairs. These pairs were then filtered based on semantic similarity, resulting in a filtered version of the dataset with 706,488 pairs.

TAT (Alkurdi et al., 2022) TAT is another paraphrasing dataset created using the same methodology as OST. The initial parallel corpus originates from Tatoeba[31]. The unfiltered and filtered versions of the dataset include 265,203 and 50,423 pairs, respectively.

TR-News (Baykara and Güngör, 2022) TR-News is a collection of news articles along with corresponding summaries and titles covering a wide range of topics. It is compiled from three Turkish national news outlets: Cumhuriyet, NTV, and HaberTürk. The dataset consists of approximately 307K articles, split into 277,573 train, 14,610 validation, and 15,379 test documents.

MLSUM (Scialom et al., 2020) MLSUM is a large-scale, multilingual summarization dataset that includes Turkish articles. The Turkish subset contains 273,617 articles from InternetHaber, further divided into 259,277 train, 11,565 validation, and 12,755 test documents.

WikiANN (Rahimi et al., 2019) WikiANN is a multilingual named entity recognition dataset containing instances from Wikipedia articles annotated with tags of "location", "person", and "organization". The Turkish subset of the dataset includes 40,000 rows, split into 20,000 for training, 10,000 for validation, and 10,000 for testing.

MilliyetNER (Tür et al., 2003) Milliyet NER is a named entity recognition dataset that includes instances from Turkish news articles annotated with tags of "location", "person", and "organization". The dataset comprises 515,123 words, divided into a training set of 419,996, a validation set of 45,532 and a test set of 49,595 words.

UD Turkish IMST (Türk et al., 2023) The IMST-UD Treebank is a Turkish dependency treebank in the format of the Universal Dependencies (UD) framework (Sulubacak and Eryiğit, 2018). The treebank was annotated manually in a format other than UD, and then automatically converted for the UD version v1.3 to be the first Turkish UD treebank. It has since then received various updates and corrections. The latest version, v2.13, has 56,422 tokens in total, with 36,415 tokens for training, 10,257 for validation, and 9,750 for testing.

UD Turkish BOUN (Marşan et al., 2023) The BOUN treebank is another Turkish dependency treebank that has been a part of the UD project since v2.7. Since then, it has received a few updates with corrections. The latest version, v2.13, has 121,835 tokens in total, with 97,797 tokens for training, 12,023 for validation, and 12,015 for testing.

STSb-TR (Beken Fikri et al., 2021) STSb-TR is derived from the English Semantic Textual Similarity benchmark (STSb) dataset (Cer et al., 2017) by translating the English sentences into Turkish using Google Translate, with no manual corrections. Each data element has two sentences and a corresponding similarity score. The dataset contains 5,749 training, 1,500 validation and 1,379 test samples.

NLI-TR (Budur et al., 2020) The Natural Language Inference in Turkish (NLI-TR) dataset consists of two large-scale datasets containing pairs of sentences labeled as "entailment", "contradiction", or "neutral". These sentence pairs were obtained by translating the widely used NLI corpora, made up of SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018). The SNLI dataset includes 570K samples, with 550K for training, 10K for validation, and 10K for testing. The MultiNLI dataset contains 413K samples, with 393K for training and 20K for validation, evenly divided between matched and mismatched pairs.

Product Reviews The Turkish Product Reviews is a sentiment classification dataset that contains product reviews from various online sources, and is available on Hugging Face[32]. A total of 235,165 reviews are categorized as positive or negative. We deduplicated the dataset before usage, and split it with an 80-10-10 train-validation-test ratio. The resulting dataset contains 186,806 training, 23,351 validation and 23,351 test samples.

TTC4900 (Yıldırım and Yıldız, 2018) The dataset is made available by the Kemik NLP

[31] tatoeba.org

[32] hf.co/datasets/turkish_product_reviews

Group[33], and contains 4,900 news articles and texts classified with one of seven categories: economy, culture-arts, health, politics, sports, technology and world. The dataset is available on Kaggle[34] and Hugging Face[35]. The TTC4900 data was also deduplicated before fine-tuning, and split with an 80-10-10 ratio, leaving 3,631 samples for training, and 454 samples each for test and validation.

Tweet Sentiments (Amasyali et al., 2018) Tweet Sentiments is a sentiment classification dataset with three categories: positive, negative and neutral. The dataset consists of 17,289 tweets that contain comments about a GSM operator, split into 13,832 training and 3,457 test samples. Due to lack of a validation set, the training set was split with a 90-10 train-validation ratio. After deduplication, the resulting fine-tuning dataset contains 12,421 training, 1,381 validation and 3,456 test samples.

A.3 Fine-tuning details

Data splits. We used predefined splits for datasets, including training, validation, and test sets. For datasets lacking both validation and test sets, we divided the data into training, validation, and test sets with an 80-10-10 ratio. In the absence of the validation set only, we utilized 10% of the original training data to generate a validation set, while the remaining 90% was used for training. We used the same approach for datasets that lacked a test set. For the NLI task, we fine-tuned our model on the training set referred to as NLI-TR (Budur et al., 2020), which is the combination of the training sets of SNLI-TR and MultiNLI-TR, and we used the already existing test and validation sets of the SNLI-TR dataset.

Dataset-specific parameters. Considering the varying lengths of dataset samples, we used dataset-specific parameters. These parameters set the maximum input and target lengths, and batch size to fit into the largest batch. In order to speed up the fine-tuning process, we employed bf16 mixed precision in the summarization and title generation experiments, allowing for a larger batch size. Table 7 shows the hyperparameters used for fine-tuning.

A.4 Mode-Switching

In the UL2 framework, specific sentinel tokens are dedicated to different pretraining objectives, enabling the model to adjust its mode for optimal task performance. This approach is also applied to fine-tuning and few-shot learning by using a token tailored to the needs of the downstream task, such as `[S2S]` for generation tasks. This is known as mode switching.

We tested mode switching by fine-tuning TURNA on several tasks and datasets. The results, detailed in Tables 8, 9, and 10, showed that TURNA models fine-tuned without any sentinel token scored highest on paraphrasing evaluations. However, a separate sentinel token achieved the best scores on different classification datasets, with the scores being remarkably close. In the semantic textual similarity task, the model trained with the `[NLG]` token performed the best.

We found no consistent pattern in the performance of different tokens across various tasks and datasets. This suggests that mode-switching might not always enhance performance, and could potentially degrade it.

Table 9: Comparison of mode switching modes on the text classification task.

Dataset	Mode	Precision	Recall	F1
Product Reviews	-	94.67	95.24	94.81
	[NLG]	94.30	95.03	94.39
	[NLU]	94.45	95.10	94.60
	[S2S]	94.34	95.04	94.47
TTC4900	-	89.15	88.11	88.16
	[NLG]	89.50	88.33	88.39
	[NLU]	86.18	84.14	84.31
	[S2S]	90.83	90.31	90.24
Tweet Sentiment	-	74.58	73.78	73.94
	[NLG]	76.01	75.84	75.56
	[NLU]	75.45	75.46	75.45
	[S2S]	75.55	74.91	74.86

Table 10: Comparison of mode switching modes on semantic textual similarity (STS).

Mode	Pearson
-	78.74
[NLG]	79.71
[NLU]	78.45
[S2S]	78.30

A.5 Hyperparameter Tuning

We conducted an additional experiment on the Semantic Textual Similarity task due to the low

[33] kemik.yildiz.edu.tr
[34] kaggle.com/savasy/ttc4900
[35] hf.co/datasets/ttc4900

Table 7: Dataset-specific hyperparameters for fine-tuning

Task	**Dataset**	**Max Input Length**	**Max Target Length**	**Batch Size**
Summarization	TRNews	768	128	4
	MLSUM	768	128	4
Title Generation	TRNews	256	64	8
	MLSUM	256	64	8
Paraphrasing	Tatoeba	20	20	128
	OpenSubtitles	20	20	128
NER	WikiANN	60	40	64
	MilliyetNER	380	60	8
POS	BOUN	90	300	8
	IMST	60	210	16
NLI	NLI-TR	128	8	32
Classification	Product Reviews	20	4	32
	TTC4900	1,450	8	2
	Tweet Sentiment	160	4	32
STS	STSb-TR	140	10	32

Table 8: Comparison of mode switching modes on the paraphrasing task.

Dataset	**Mode**	**Rouge1**	**Rouge2**	**RougeL**	**BLEU**	**METEOR**
OST	-	78.43	63.58	76.81	51.47	74.79
	[NLG]	76.20	61.11	74.50	46.27	73.76
	[NLU]	77.18	61.97	75.33	48.39	74.02
	[S2S]	77.20	61.98	75.44	48.53	74.05
TAT	-	90.22	80.23	88.95	71.14	87.56
	[NLG]	89.66	79.28	88.41	69.54	87.18
	[NLU]	89.08	78.53	87.90	68.33	86.82
	[S2S]	89.71	79.37	88.45	69.61	87.26

Pearson correlation score obtained by TURNA-Encoder when compared to TURNA and BERTurk. We fine-tuned TURNA-Encoder with different learning rates on the regression task. The results are reported in Table 11. The difference in Pearson correlation scores suggest that elaborate hyperparameter tuning can significantly alter the downstream performance of our model.

Table 11: Comparison of TURNA-Encoder performance with different learning rates on semantic textual similarity (STS).

Learning Rate	**Pearson**
(Default) 5×10^{-5}	73.63
5×10^{-4}	77.13
5×10^{-3}	-3.56
5×10^{-2}	17.92