FaBERT: Pre-training BERT on Persian Blogs

Mostafa Masumi[♢†], Seyed Soroush Majd[♢], Mehrnoush Shamsfard[♢], and Hamid Beigy[†]

[♢]Computer Science and Engineering Department, Shahid Beheshti University
[♢]*s.majd@mail.sbu.ac.ir, m-shams@sbu.ac.ir*
[†]Computer Engineering Department, Sharif University of Technology
[†]*{m.masumi, beigy}@sharif.edu*

Abstract

We introduce FaBERT, a Persian BERT-base model pre-trained on the HmBlogs corpus, encompassing both informal and formal Persian texts. FaBERT is designed to excel in traditional Natural Language Understanding (NLU) tasks, addressing the intricacies of diverse sentence structures and linguistic styles prevalent in the Persian language. In our comprehensive evaluation of FaBERT on 12 datasets in various downstream tasks, encompassing Sentiment Analysis (SA), Named Entity Recognition (NER), Natural Language Inference (NLI), Question Answering (QA), and Question Paraphrasing (QP), it consistently demonstrated improved performance, all achieved within a compact model size. The findings highlight the importance of utilizing diverse corpora, such as HmBlogs, to enhance the performance of language models like BERT in Persian Natural Language Processing (NLP) applications. FaBERT is openly accessible at https://huggingface.co/sbunlp/fabert.

1 Introduction

Recently, we've seen the rise of sophisticated language models like BERT (Devlin et al., 2019), transforming the understanding of languages, including Persian. Whether designed for multiple languages or specifically for Persian, these models have been employed across various applications in Persian Natural Language Processing (NLP). Their training encompassed a diverse range of textual sources, including websites like Wikipedia and social media platforms such as Twitter, as well as news articles and academic journals.

More recently, Large Language Models (LLMs) with a substantial increase in parameters have significantly reshaped the landscape of NLP, excelling in a myriad of tasks. Despite their significant contributions, finely-tuned LMs such as BERT still demonstrate robust performance, achieving comparable results or, in many cases, even outperforming LLMs in traditional Natural Language Understanding (NLU) tasks, including Natural Language Inference (NLI), Sentiment Analysis, Text Classification, and Question Answering (QA) (Yang et al., 2023). Encoder-only models like BERT remain the workhorses of practical language processing, with applications ranging from content moderation to information retrieval systems.

Additionally, LLMs often come with the drawback of slower response times and increased latency compared to smaller models. Moreover, the use of LLMs typically demands advanced hardware, creating accessibility challenges for many users. Privacy concerns may also emerge when employing LLMs online. Notably, encoder models like BERT have found crucial roles in supporting LLM deployments, serving as efficient filters for content safety (Ji et al., 2024), performing rapid document retrieval in RAG systems (Lewis et al., 2020), and enabling cost-effective preprocessing of large-scale data (Penedo et al., 2024). Their compact size and efficient architecture make them particularly suitable for edge devices and mobile applications, where computational resources and power consumption are constrained.

Recent studies (Nguyen et al., 2020; Abdelali et al., 2021) highlight the value of incorporating informal text into training corpora, as it improves a model's ability to handle colloquial language and social media content, leading to better performance on diverse linguistic tasks.

Our motivation is to develop FaBERT, a Persian BERT model exclusively pre-trained on Persian blogs, to enhance performance in traditional NLU tasks and enable efficient processing of both formal and informal texts in the language. Blogs, which have not previously been utilized for pre-

Proceedings of the Tenth Workshop on Noisy and User-generated Text, pages 85–96

training Persian LMs, serve as a rich source of colloquial language with flexible sentence structures, idiomatic expressions, and informal lexicons inherent in everyday Persian communication. While recent models have demonstrated commendable capabilities, there still remains room for improvement, particularly in tasks involving informal Persian text. Blog content includes diverse and evolving language variations such as cultural references, informal lexicons, and slang in Persian, which have been user-generated across different demographics over a long period, contributing to FaBERT's robust performance.

Our findings reveal that pre-training on the HmBlogs corpus from Persian blogs enhances the model's performance, leading to state-of-the-art results across various downstream tasks. The main contributions of this paper are:

1. Pre-training a BERT-base model on Persian blog texts in the HmBlogs corpus and making it publicly accessible.

2. Evaluating the model's performance on 12 datasets in various downstream tasks, including sentiment analysis, irony detection, natural language inference, question paraphrasing, named entity recognition, and question answering.

The subsequent sections of the paper are structured as follows: Section 2 provides an introduction and comparison of various BERT models employed for Persian NLP. Section 3 delves into the details of our corpus, model, and its pre-training procedure. Section 4 compares FaBERT's performance in downstream tasks with other models. Finally, Section 5 concludes the paper by summarizing our findings.

2 Related Work

BERT that stands as Bidirectional Encoder Representations from Transformers, has demonstrated its exceptional abilities across a wide range of natural language understanding tasks. Unlike traditional language models that process text in a unidirectional manner (left-to-right or right-to-left), BERT considers both the left and right context of words.

BERT's pre-training involved two training objectives: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). MLM randomly masks words in a sentence, and the model learns to predict the missing words based on context, enhancing its ability to grasp the semantic meaning and relationships between words within sentences. On the other hand, in the NSP task, the model has to predict whether sentence B logically succeeds sentence A. MLM and NSP are designed for the model to learn a language representation, which can then be used to extract features for downstream tasks. Continuing the discussion, we will present a selection of Persian-language BERT models.

The most well-known Persian language model is ParsBERT (Farahani et al., 2021). It was pre-trained using both MLM and NSP tasks, utilizing a training corpus collected from 8 different sources. ParsBERT has become the preferred choice for Persian NLP tasks, thanks to its outstanding performance.

AriaBERT (Ghafouri et al., 2023) is another Persian language model that follows RoBERTa's enhancements (Liu et al., 2019) and utilizes Byte-Pair Encoding tokenizer. Its diverse training dataset, exceeding 32 gigabytes, includes conversational, formal, and hybrid texts.

Additionally, many Multilingual Language Models have been released since, and few of them include Persian. Multilingual BERT, also known as mBERT, was introduced by Devlin et al. (2019). It was trained with NSP and MLM tasks on the Wikipedia pages of 104 languages with a shared word-piece vocabulary. mBERT has shown impressive zero-shot cross-lingual transfer and is effective in utilizing task-specific annotations from one language for fine-tuning and evaluation in another. Although mBERT has shown solid performance across different languages, monolingual BERT models outperform mBERT in most downstream tasks.

Similarly, XLM-R (Conneau et al., 2019), an extension of the RoBERTa model by Facebook AI, is designed for cross-lingual understanding. This model was pre-trained with the MLM objective on a vast corpus comprising more than 2 terabytes of text from 100 languages and outperformed mBERT in many downstream tasks.

The models previously reviewed adhere to the architecture introduced by the original BERT-base model, featuring 12 layers and 12 attention heads. While maintaining this consistency, there are variations in vocabulary size among these models.

A larger vocabulary facilitates the capture of more unique tokens and their relationships, but it

comes at the expense of an increased number of parameters. This, in turn, necessitates more extensive training data for learning embeddings. Conversely, smaller vocabularies may struggle to capture all the details of language, potentially causing information and context to be lost. An instance is found in the multilingual model mBERT, which supports 100 different languages with a vocabulary size of only 100,000. Despite the broad language coverage, this choice leads to a limited set of tokens for each language. Consequently, sentences are transformed into a greater number of tokens, potentially exceeding the maximum supported sequence length and resulting in the loss of information. Table 1 summarizes the vocabulary size and number of parameters for each model under consideration.

Model	Vocabulary Size (K)	# of Parameters (M)
BERT (English)	30	109
mBERT	105	167
XLM-R	250	278
ParsBERT	100	162
AriaBERT	60	132
FaBERT	50	124

Table 1: Vocabulary Size and Parameter Count of Persian BERT Models

3 Methodology

3.1 Training Corpus

The selection of an appropriate training corpus is a pivotal element in the pre-training of a language model. For this effort, we utilized the HmBlogs corpus (Khansari and Shamsfard, 2021), a collection of 20 million posts of Persian blogs over 15 years. HmBlogs includes more than 6.8 billion tokens, covering a wide range of topics, genres, and writing styles, including both formal and informal texts together.

To ensure high-quality pre-training, a series of pre-processing steps were performed on the corpus. Many posts written in the Persian alphabet were erroneously identified as Persian despite not being in the Persian language. This confusion arises from the Persian alphabet's resemblance to the alphabets of other languages like Arabic and Kurdish. Additionally, some other posts had typographical errors, very rare words, or the excessive use of local dialects. Therefore, a post-discriminator was implemented to filter out these improper and noisy posts.

Cleaning documents in Persian poses another challenge due to the presence of non-standard characters[1]. These characters look identical to Persian characters, but their different codes can cause problems during pre-training. Some Persian blogs may also use decorative characters to make the text visually appealing. Such characters were standardized to ensure uniform representation and avoid potential discrepancies. Additionally, words with repetitive characters were corrected.

3.2 Pre-training Procedure

We trained a BERT model following the architecture proposed by Devlin et al. (2019). Our BERT-base model, FaBERT, adheres to the original BERT-base architecture, consisting of 12 hidden layers, each with 12 self-attention heads.

We opted for the WordPiece tokenizer over alternatives such as BPE, as prior evidence indicates no performance improvement (Geiping and Goldstein, 2023), and with a conservative stance, we set the vocabulary size to 50,000 tokens. This decision aimed at finding a balance between capturing linguistic details and managing the computational demands associated with larger vocabularies. It's essential to note that Persian text includes half spaces, a feature absent in English. Consequently, the FaBERT tokenizer has been adapted to handle this feature, ensuring appropriate representation of texts during pre-training and fine-tuning.

The total number of parameters for FaBERT is 124 million. In comparison to other Persian and multilingual base models outlined in Table 1, FaBERT is more compact with fewer parameters.

During pre-training, each input consisted of one or more sentences sampled contiguously from a single document. The samples were of varying lengths to help the model effectively learn the positional encodings.

We implemented dynamic masking, inspired by the methodology introduced by Liu et al. (2019), and omitted the Next Sentence Prediction task from our pre-training process, as it was demonstrated to have no discernible positive impact on performance. The masking rate for dynamic masking was set to 15%. We also utilized the whole word masking approach for enhanced performance. Unlike traditional MLM, which randomly masks individual tokens in a sentence, whole word masking involves

[1]For instance, Arabic 'ي' and 'ك' are occasionally substituted for Persian 'ی' and 'ک'.

Hyperparameter	Value	Hyperparameter	Value
Batch Size	32	Total Steps	18 Million
Optimizer	Adam	Warmup Steps	1.8 Million
Learning Rate	6e-5	Precision Format	TF32
Weight Decay	0.01	Dropout	0.1

Table 2: Pre-training Hyperparameters

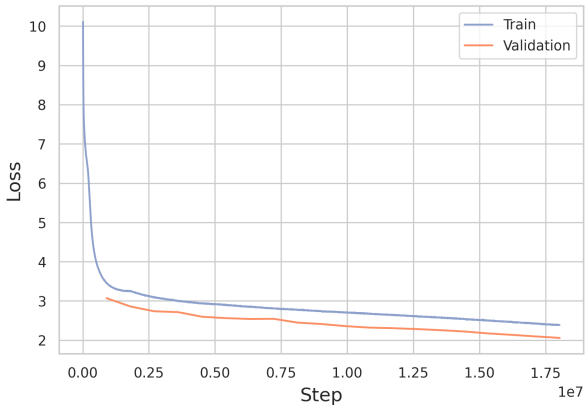

Figure 1: Train and Validation MLM loss in pre-training

masking entire words. Table 2 details the hyperparameters used in the pre-training process.

The training was conducted on a single Nvidia A100 40GB GPU, spanning a duration of 400 hours. The training data was split into 99% for training and 1% for validation. The final validation perplexity achieved was 7.76, and the train and validation loss plot is presented in Figure 1.

4 Experiments and Results

In this section, we assess the FaBERT model across four different categories of downstream tasks. For NLI and Question Paraphrasing, sentence pairs are processed to generate labels based on their relationship. In NER, entities within single input sentences are labeled at the token level. Sentiment Analysis and Irony Detection involve processing individual sentences and assigning corresponding labels. In Question Answering, models utilize a given question and the provided paragraph to generate token-level spans for answers.

For each task, we fine-tuned FaBERT and compared its performance to other state-of-the-art models, such as ParsBERT (Farahani et al., 2021), AriaBERT (Ghafouri et al., 2023), and multilingual models like mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2019). Lastly, we analyze the effectiveness of FaBERT's tokenizer and compare it with other BERT models.

To ensure a fair comparison, all models were fine-tuned on the same datasets using consistent train/validation/test splits. For each model and dataset pair, we performed a grid search over hyperparameters, selecting the configuration that achieved the best validation score. The scores reported in this paper correspond to the test set results obtained under these optimal conditions. Details of the grid search ranges and dataset splits are provided in Appendix A.

4.1 Natural Language Inference and Question Paraphrasing

In this section, we analyze FaBERT's ability to understand logical and semantic relationships between sentences, focusing on tasks like Natural NLI and Question Paraphrasing. We assess its performance using the Farstail (Amirkhani et al., 2023), SBU-NLI (Rahimi and ShamsFard, 2024), and ParsiNLU Question Paraphrasing (Khashabi et al., 2021) datasets.

FarsTail

The FarsTail NLI dataset is sourced from multiple-choice questions from various subjects, specifically collected from Iranian university exams. Each of these questions became the basis for generating NLI instances with three different relationships: Entailment, Contradiction, and Neutral.

SBU-NLI

SBU-NLI is another dataset containing sentence pairs categorized into three labels: Entailment, Contradiction, and Neutral. This data is gathered from various sources to create a balanced dataset.

ParsiNLU Question Paraphrasing

This task involves determining the relationship between pairs of questions, specifically classifying whether they are paraphrases. The dataset is created through two means: first, by mining questions from Google auto-complete and Persian discussion forums, and second, by translating the QQP dataset with Google Translate API. As a result, some questions are presented in an informal fashion.

As observed in Table 3, FaBERT demonstrates a +1% improvement in F1 for FarsTail, comparable performance to mBERT in SBU-NLI, and a +2.88% F1 score in the informal ParsiNLU Question Paraphrasing dataset.

4.2 Named Entity Recognition

In this section, we assess the efficacy of FaBERT in NER, a commonly employed intermediate task that

Model	FarsTail	SBU-NLI	Parsi-NLU QP
ParsBERT	82.52	58.41	77.60
mBERT	83.42	66.38	79.48
XLM-R	83.50	58.85	79.74
AriaBERT	76.39	52.81	78.86
FaBERT	**84.45**	**66.65**	**82.62**

Table 3: Performance Comparison in NLI and Question Paraphrasing

Model	ParsTwiner	PEYMA	MultiCoNER v2
ParsBERT	81.13	91.24	**58.09**
mBERT	75.60	87.84	51.04
XLM-R	79.50	90.91	51.47
AriaBERT	78.53	89.76	54.00
FaBERT	**82.22**	**91.39**	57.92

Table 4: Performance Comparison in Named Entity Recognition

facilitates information extraction and entity identification within textual data. Our assessment leveraged formal and informal datasets, including ParsTwiNER (Aghajani et al., 2021), PEYMA (Shahshahani et al., 2018), and MultiCoNER v2 (Fetahu et al., 2023). The comparison of different models for each entity type is detailed in Appendix B.

ParsTwiNER

The ParsTwiNER offers a NER dataset gathered from 7632 tweets collected from the Persian Twitter accounts, offering diverse informal Persian content. Annotation by experts in natural language processing resulted in 24061 named entities across categories such as persons, organizations, locations, events, groups, and nations.

PEYMA

The PEYMA NER dataset, derived from formal text extracted from ten news websites, classifies words into different categories, encompassing persons, locations, organizations, time, date, and more. PEYMA is known as a key asset for training and evaluating NER systems in the Persian language.

MultiCoNER v2

Initially introduced as a part of SemEval task in 2022, MultiCoNER is a multilingual NER dataset crafted to address contemporary challenges in NER, such as low-context scenarios, syntactically complex entities like movie titles, and long-tail entity distributions. The enhanced version of this dataset was used in the following year as part of the SemEval 2023 task. This version, known as MultiCoNER v2, expanded these challenges by adding fine-grained entities and inserting noise in the input text. Gathered from Wikidata and Wikipedia, the dataset spans 12 languages, with Persian being the focus of our evaluations.

The evaluation metrics used include micro-F1 for PEYMA and ParsTwiNER datasets, and macro-F1 for MultiCoNER v2. Table 4 provides a detailed overview of scores achieved by each model. Across the board, all models demonstrated comparable performance in the PEYMA dataset. However, FaBERT model exhibited a slight improvement by achieving a +1.09% increase in F1 score for the informal ParsTwiNER dataset. In the MultiCoNER v2 dataset, both FaBERT and ParsBERT outperformed other models. In general FaBERT and ParsBERT seem to be great options for applications involving NER.

4.3 Sentiment Analysis and Irony Detection

In this section, we assess FaBERT's performance in classifying expressions. We employed DeepSentiPers (Sharami et al., 2020), MirasOpinion (Asli et al., 2020), and MirasIrony (Golazizian et al., 2020) datasets for evaluation.

DeepSentiPers

The DeepSentiPers dataset comprises 9,000 customer reviews of Digikala, an Iranian E-commerce platform. Originally, each sentence's polarity was annotated using a 5-class label set $E = \{-2, -1, 0, +1, +2\}$, representing sentiments from very displeased to delighted. However, our investigation revealed inconsistencies, particularly between the -1 and -2 categories for negative sentiments and the +1 and +2 categories for positive sentiments. Recognizing the overlap between these closely related labels, we opted for a simplified 3-class labeling approach, classifying sentiments as negative, neutral, or positive.

MirasOpinion

MirasOpinion, the largest Persian Sentiment dataset, comprises 93,000 reviews gathered from the Digikala platform. Through crowdsourcing, each review was labeled as Positive, Neutral, or Negative. This dataset was included in the SPARROW, a benchmark for sociopragmatic meaning understanding. Participating in the SPARROW

benchmark (Zhang et al., 2023) allowed us to assess FaBERT against various language models.

MirasIrony

MirasIrony, a 2-labeled dataset designed for irony detection, encompasses 4,339 manually labeled Persian tweets. In this dataset, tweets exhibiting a disparity between their literal meaning and sentiment were labeled as positive, while those lacking this characteristic were labeled as negative. Similar to MirasOpinion, we assessed the performance of models on MirasIrony using the SPARROW benchmark.

Model	DeepSentiPers	MirasOpinion	MirasIrony
ParsBERT	74.94	86.73	71.08
mBERT	72.95	84.40	74.48
XLM-R	79.00	84.92	**75.51**
AriaBERT	75.09	85.56	73.80
FaBERT	**79.85**	**87.51**	74.82

Table 5: Performance Comparison in Sentiment Analysis and Irony Detection

Macro averaged F1 score serves as the evaluation metric for DeepSentiPers and MirasOpinion, while Accuracy is employed for MirasIrony. As presented in Table 5, FaBERT achieved the highest scores in sentiment analysis for both DeepSentiPers and MirasOpinion. For irony detection in the MirasIrony dataset, XLM-R outperforms other models, securing the leading position with a score of 75.51%. FaBERT demonstrated notable performance as well, securing the second spot with 74.82% accuracy. Through the SPARROW benchmark leaderboard, other models can be compared with FaBERT on MirasOpinion[2] and MirasIrony[3] tasks.

4.4 Question Answering

To evaluate the question-answering capabilities of FaBERT, our experiments encompassed three datasets: ParsiNLU Reading Comprehension (Khashabi et al., 2021), PQuad (Darvishi et al., 2023), and PCoQA (Hemati et al., 2023). Each dataset is briefly introduced in the following sections. Table 6 summarizes the performance of different models on each dataset.

[2] https://sparrow.dlnlp.ai/sentiment-2020-ashrafi-fas.taskshow

[3] https://sparrow.dlnlp.ai/irony-2020-golazizian-fas.taskshow

ParsiNLU Reading Comprehension Dataset

Reading Comprehension is one of the tasks introduced in the ParsiNLU benchmark and involves extracting a substring from a given context paragraph to answer a specific question. In order to create this dataset, they used Google's Autocomplete API to mine questions deemed popular by users. Starting with a seed set of questions, they repeatedly queried previous questions to expand on the set and add more sophisticated ones. After filtering out invalid questions, native annotators then chose the pertinent text span from relevant paragraphs that provided the answer to each question.

The evaluation of models on this dataset involves comparing the answers generated by the models to the provided ground truth answers. The main metrics used are the F1 score, which measures the overlap between the predicted and ground truth answers, and the exact match (EM) score, which checks if the predicted answers exactly match the ground truth answers. FaBERT scored +6.24% higher in F1 compared to other models in the ParsiNLU Reading Comprehension task.

PQuAD: A Persian question answering dataset

PQuAD is a large-scale, human-annotated question-answering dataset for the Persian language. It contains 80,000 questions based on passages extracted from Persian Wikipedia articles. The questions and their corresponding answers were generated through a crowdsourcing process, where crowdworkers were presented with passages and tasked with crafting questions and corresponding answers based on the provided content. Inspired by the structure of SQuAD 2.0 (Rajpurkar et al., 2018), PQuAD designates 25% of its questions as unanswerable, adding extra complexity to the dataset and enhancing the evaluative challenge.

In this dataset, in addition to F1 and EM scores, the evaluation can be broken down into subsets of questions that have answers (HasAns) and those that do not have answers (NoAns). By considering these metrics, the performance of different models can be compared and analyzed to determine their effectiveness in answering questions or abstaining from answering. The authors also provided an estimation of human performance by asking a group of crowdworkers to answer a subset of questions. Both FaBERT and XLM-R demonstrate remarkable capabilities in question answering, achieving a comparable F1 score performance. However, XLM-R slightly outperforms FaBERT in this aspect.

Model	ParsiNLU		PQuAD					PCoQA				
	Exact Match	F1	Exact Match	F1	HasAns EM	HasAns F1	NoAns	Exact Match	F1	HEQ-Q	HEQ-M	NoAns
ParsBERT	22.10	44.89	74.41	86.89	68.97	85.34	91.79	31.17	50.96	41.07	0.81	48.83
mBERT	26.31	49.63	73.68	86.71	67.52	84.66	**93.26**	26.89	46.11	36.94	1.63	31.62
XLM-R	21.92	42.55	**75.16**	**87.60**	69.79	86.13	92.26	34.52	51.12	44.81	0.81	54.88
AriaBERT	16.49	37.98	69.70	82.71	63.61	80.71	89.08	22.68	41.37	32.89	0	40.93
FaBERT	**33.33**	**55.87**	75.04	87.34	**70.33**	**86.50**	90.02	**35.85**	**53.51**	**45.36**	**2.45**	**61.39**
Human	-	-	80.3	88.3	74.9	85.6	96.80	85.5	86.97	-	-	-

Table 6: Performance Comparison in Question Answering

PCoQA: Persian Conversational Question Answering Dataset

PCoQA is the first dataset designed for answering conversational questions in Persian. It comprises 870 dialogs and over 9,000 question-answer pairs sourced from Wikipedia articles. In this task, contextually connected questions are posed about a given document, and models are required to respond by extracting relevant information from given paragraphs. This dataset provides a suitable context for assessing the model's performance in Persian conversational question answering, similar to the English dataset CoQA (Reddy et al., 2019).

For the PCoQA dataset, in addition to F1 and EM scores, two variants of human equivalence score (HEQ) are suggested by the authors. HEQ-Q measures the percentage of questions for which system F1 exceeds or matches human F1, and HEQ-M quantifies the number of dialogs for which the model achieves a better overall performance compared to the human. FaBERT outperformed other models with +2.39% higher F1 score, handling both answerable and unanswerable questions well. Additionally, the PCoQA dataset proves to be challenging, with all models scoring noticeably lower than humans.

4.5 Vocabulary Impact on Input Length

To evaluate the impact of FaBERT's chosen vocabulary size on its effective maximum input length, a comparative analysis was conducted across datasets with longer sentences, including MirasOpinion, FarsTail, ParsiNLU Reading Comprehension, and PQuAD. The objective was to examine how different tokenizers, including the one trained for FaBERT, influence the number of tokens in each input sentence.

Table 7 provides a summary of median token counts across the aforementioned datasets. Both multilingual models faced challenges due to the lack of sufficient Persian tokens in their vocabularies, potentially impacting their performance on longer inputs due to loss of information. ParsBERT's tokenizer yields the most compact sequences, closely followed by FaBERT. An interesting observation arises in the PQuAD dataset, where ParsBERT outperforms, likely attributed to PQuAD's reliance on Wikipedia, a significant component of ParsBERT's pre-training data.

Overall, FaBERT's tokenizer, despite having a vocabulary size half that of ParsBERT, demonstrated a comparable level of compression. The detailed boxplots for each dataset are available in Appendix C.

Tokenizer	MirasOpinion	FarsTail	ParsiNLU RC	PQuAD
ParsBERT	27	58	113.5	160
mBERT	44	85	165	235
XLM-R	34	74	142.5	210
AriaBERT	28	66	130	207
FaBERT	28	62	119.5	189

Table 7: Median Token Count Yielded by Different Tokenizers

5 Conclusion

In this paper, we pre-trained FaBERT, a BERT-base model from scratch exclusively on the diverse HmBlogs corpus, consisting solely of raw texts from Persian blogs. Notably, our model's smaller vocabulary size resulted in a more compact overall size compared to competitors. FaBERT performed exceptionally well in 12 different datasets, outperforming competitors in nine of them. In the remaining tasks where it did not secure the top position, it consistently ranked among the top performers, closely following the highest-performing model.

Our results indicate that texts with diverse writing styles, both formal and informal, found in Persian blogs can significantly contribute to the high-quality pre-training of language models, including BERT. The effectiveness of the Hmblogs corpus in the performance of our BERT model in downstream tasks demonstrates its potential for being used in pre-training both language models and

large language models alongside other relevant Persian corpora. This success aligns with the broader trend in NLP where encoder-only models continue to prove their value, particularly in scenarios requiring efficient processing of large-scale text data while maintaining high performance standards.

The practical advantages of our approach – combining the efficiency of BERT's architecture with rich, diverse training data – position FaBERT as a valuable tool for Persian NLP applications, especially in resource-constrained environments where larger models may be impractical. This work not only advances Persian language processing capabilities but also reinforces the continuing relevance of carefully designed encoder models in the evolving landscape of natural language processing.

Limitations

Biases As FaBERT is trained exclusively on blog data, it inherits potential demographic and sociolinguistic biases present in Persian online communities.

Technical Constraints FaBERT, like other BERT-based architectures, is limited by the standard 512-token sequence length, which impacts its ability to process longer documents or capture long-range dependencies. While our analysis in Section 4.5 shows that FaBERT's tokenizer achieves good compression for Persian text, this architectural constraint remains a challenge. Recent innovations in transformer models have successfully addressed long-context limitations (Zhang et al., 2024), and these advancements could be adapted to Persian NLP tasks in future research.

Embedding Capabilities The Persian NLP landscape faces a scarcity of datasets and benchmarks for training and evaluating text embeddings. Although contrastive learning has demonstrated success in producing high-quality sentence embeddings for other languages, the absence of Persian-specific parallel texts and semantic similarity datasets limits progress in developing such models for Persian. This gap needs to be addressed, given the increasing importance of dense retrievers and semantic search in NLP. Future efforts should prioritize creating resources tailored for Persian sentence embeddings to advance applications such as information retrieval and semantic similarity.

References

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations.

MohammadMahdi Aghajani, AliAkbar Badri, and Hamid Beigy. 2021. ParsTwiNER: A corpus for named entity recognition at informal persian. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 131–136.

Hossein Amirkhani, Mohammad AzariJafari, Soroush Faridan-Jahromi, Zeinab Kouhkan, Zohreh Pourjafari, and Azadeh Amirak. 2023. FarsTail: A persian natural language inference dataset. *Soft Computing*, pages 1–13.

Seyed Arad Ashrafi Asli, Behnam Sabeti, Zahra Majdabadi, Preni Golazizian, Reza Fahmi, and Omid Momenzadeh. 2020. Optimizing annotation effort using active learning strategies: A sentiment analysis case study in persian. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2855–2861.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Kasra Darvishi, Newsha Shahbodaghkhan, Zahra Abbasiantaeb, and Saeedeh Momtazi. 2023. PQuAD: A persian question answering dataset. *Computer Speech & Language*, 80:101486.

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.

Mehrdad Farahani, Mohammad Gharachorloo, Marzieh Farahani, and Mohammad Manthouri. 2021. ParsBert: Transformer-based model for persian language understanding. *Neural Processing Letters*, 53:3831–3847.

Besnik Fetahu, Zhiyu Chen, Sudipta Kar, Oleg Rokhlenko, and Shervin Malmasi. 2023. MultiCoNER v2: A large multilingual dataset for fine-grained and noisy named entity recognition. *arXiv preprint arXiv:2310.13213*.

Jonas Geiping and Tom Goldstein. 2023. Cramming: Training a language model on a single gpu in one day. In *International Conference on Machine Learning*, pages 11117–11143. PMLR.

Arash Ghafouri, Mohammad Amin Abbasi, and Hassan Naderi. 2023. AriaBERT: A pre-trained persian bert model for natural language understanding.

Preni Golazizian, Behnam Sabeti, Seyed Arad Ashrafi Asli, Zahra Majdabadi, Omid Momenzadeh, and Reza Fahmi. 2020. Irony detection in persian language: A transfer learning approach using emoji prediction. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2839–2845.

Hamed Hematian Hemati, Atousa Toghyani, Atena Souri, Sayed Hesam Alavian, Hossein Sameti, and Hamid Beigy. 2023. PCoQA: Persian conversational question answering dataset. *arXiv preprint arXiv:2312.04362*.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2024. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36.

Hamzeh Motahari Khansari and Mehrnoush Shamsfard. 2021. HmBlogs: A big general persian corpus. *arXiv preprint arXiv:2111.02362*.

Daniel Khashabi, Arman Cohan, Siamak Shakeri, Pedram Hosseini, Pouya Pezeshkpour, Malihe Alikhani, Moin Aminnaseri, Marzieh Bitaab, Faeze Brahman, Sarik Ghazarian, et al. 2021. ParsiNLU: A suite of language understanding challenges for persian. *Transactions of the Association for Computational Linguistics*, 9:1147–1162.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*.

Zeinab Rahimi and Mehrnoush ShamsFard. 2024. A knowledge-based approach for recognizing textual entailments with a focus on causality and contradiction. *Available at SSRN 4526759*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Mahsa Sadat Shahshahani, Mahdi Mohseni, Azadeh Shakery, and Heshaam Faili. 2018. PEYMA: A tagged corpus for persian named entities. *arXiv preprint arXiv:1801.09936*.

Javad PourMostafa Roshan Sharami, Parsa Abbasi Sarabestani, and Seyed Abolghasem Mirroshandel. 2020. Deepsentipers: Novel deep learning models trained over proposed augmented persian sentiment corpus. *arXiv preprint arXiv:2004.05328*.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. 2023. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *arXiv preprint arXiv:2304.13712*.

Chiyu Zhang, Khai Duy Doan, Qisheng Liao, and Muhammad Abdul-Mageed. 2023. The skipped beat: A study of sociopragmatic understanding in llms for 64 languages. *arXiv preprint arXiv:2310.14557*.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, et al. 2024. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1393–1412.

Appendix For "FaBERT: Pre-training BERT on Persian Blogs"

A Fine-tuning Hyperparameters

The hyperparameters employed for fine-tuning the models on each dataset, along with the respective train/validation/test split sizes, are outlined in Table 8. For the ParsiNLU benchmark, we adhered to the predefined hyperparameters in the ParsiNLU source code.

B Detailed NER Results

Tables 9, 10, and 11 present F1 scores for entities in PEYMA, MultiCoNER v2, and ParsTwiNER datasets, providing a model comparison for each entity. For instance, In MultiCoNER v2, FaBERT excels in recognizing medical entities, and ParsBERT is better at identifying creative works.

C Tokenizer Comparison Figures

Figures 2, 3, 4, and 5 illustrate the distribution of token counts for each model's tokenizer across the following datasets: PQuAD, ParsiNLU Reading Comprehension, MirasOpinion, and FarsTail. These boxplots provide a visual representation of the variation in token counts for each model.

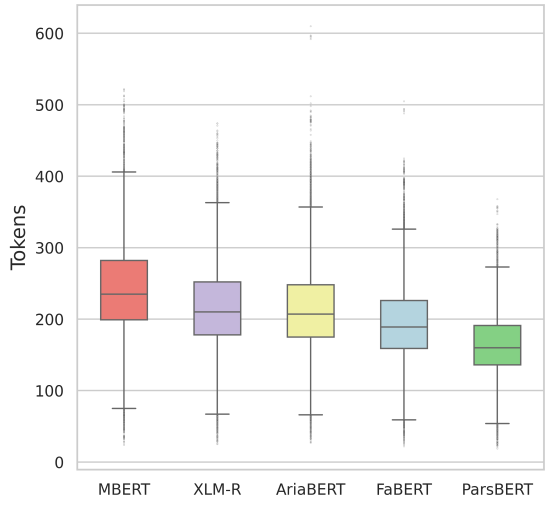

Figure 2: Token count distribution across tokenizers for the PQuAD dataset

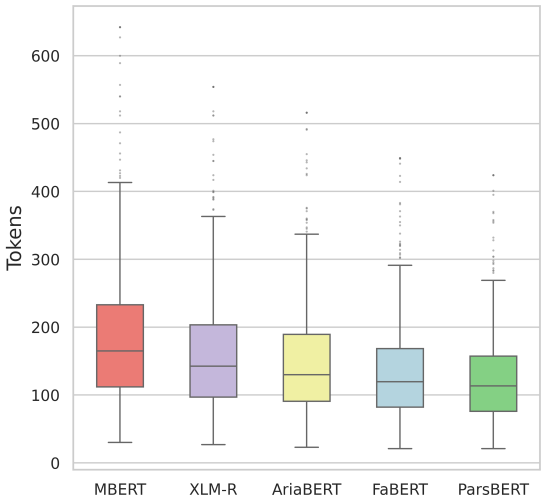

Figure 3: Token count distribution across model tokenizers for the ParsiNLU Reading Comprehension dataset

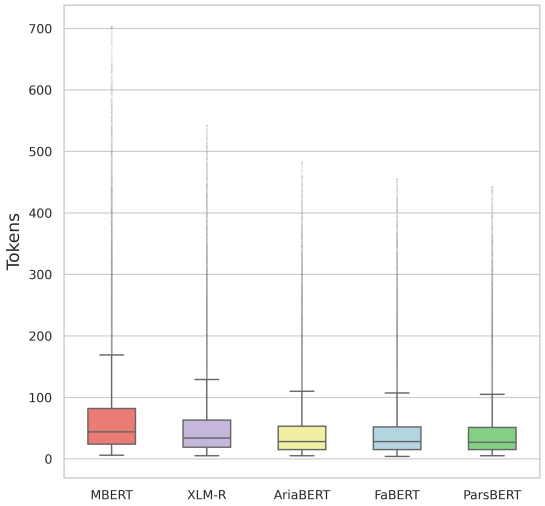

Figure 4: Token count distribution across tokenizers for the MirasOpinion dataset

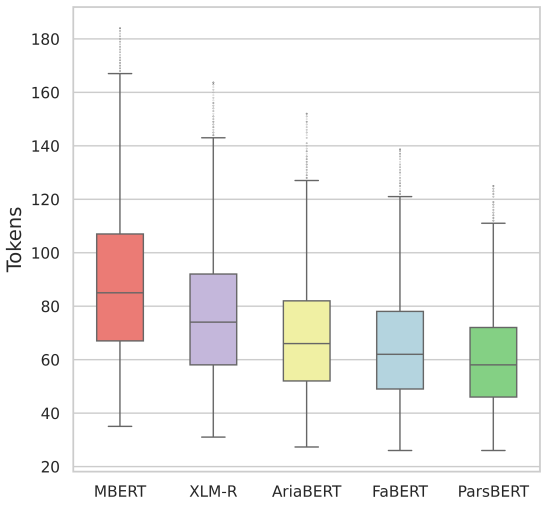

Figure 5: Token count distribution across tokenizers for the FarsTail dataset

Datasets	Train	Validation	Test	Number of Labels	Metrics	Learning Rate	Batch Size	Epochs	Warmup
DeepSentiPers	6320	703	1854	3	Macro F1	2e-5, 3e-5, 5e-5	8,16	3, 7	0, 0.2
MirasOpinion	75094	9387	9387	3	Macro F1	2e-5, 3e-5, 5e-5	8,16	1	0, 0.2
MirasIrony	2352	295	294	2	Accuracy	2e-5, 3e-5, 5e-5	8,16	3, 5	0, 0.2
PQuAD	63994	7976	8002	-	Micro F1	2e-5, 3e-5, 5e-5	8,16	2	0, 0.2
PCoQA	6319	1354	1354	-	Micro F1	3e-5, 5e-5	8,16	3, 7	0, 0.2
ParsiNLU RC	600	125	575	-	Micro F1	3e-5, 5e-5	4	3, 7	0
SBU-NLI	3248	361	401	3	Micro F1	2e-5, 3e-5, 5e-5	8,16	3, 7	0, 0.2
FarsTail	7266	1564	1537	3	Micro F1	2e-5, 3e-5, 5e-5	8,16	3, 7	0, 0.2
ParsiNLU QP	1830	898	1916	2	Micro F1	3e-5, 5e-5	8,16	3, 7	0
PEYMA	8029	926	1027	-	Macro F1	2e-5, 3e-5, 5e-5	8,16	3, 7	0, 0.2
MultiCoNER v2	16321	855	219168	-	Micro F1	2e-5, 3e-5, 5e-5	8,16	3, 7	0, 0.2
ParsTwiNER	6418	447	304	-	Micro F1	2e-5, 3e-5, 5e-5	8,16	3, 7	0, 0.2

Table 8: Dataset Split Sizes and Fine-Tuning Hyperparameters

Entity Type	FaBERT	ParsBERT	AriaBERT	mBERT	XLM-R	Support
Date	89.16	85.65	85.11	84.56	86.73	208
Location	91.95	91.73	91.46	90.25	92.42	595
Currency	94.34	94.34	83.64	90.57	96.15	26
Organization	88.24	89.37	86.38	84.83	87.25	667
Percent	98.63	98.63	93.33	97.14	94.74	36
Person	95.45	95.29	94.6	90.1	95.75	434
Time	96.97	91.43	96.97	76.47	94.12	16
Micro Average	91.39	91.24	89.76	87.84	90.91	1982
Macro Average	93.53	92.35	90.21	87.7	92.45	1982
Weighted Average	91.37	91.23	89.75	87.81	90.92	1982

Table 9: Comparison of F1 Scores for Each Entity Type in PEYMA

Entity Type	FaBERT	ParsBERT	AriaBERT	mBERT	XLM-R	Support
Event	0.5714	0.4444	0.4118	0.4865	0.2308	14
Location	0.8281	0.8414	0.7991	0.7802	0.8088	221
Nation	0.9	0.7385	0.7246	0.7123	0.7397	30
Organization	0.7364	0.6966	0.6691	0.6462	0.7126	129
Person	0.9344	0.8893	0.8745	0.8216	0.8629	244
Political Group	0.6364	0.6667	0.7442	0.7	0.8	22
Micro Average	0.8222	0.8113	0.7853	0.756	0.795	660
Macro Average	0.7301	0.7128	0.7039	0.6911	0.6925	660
Weighted Average	0.8238	0.8119	0.7881	0.7573	0.7943	660

Table 10: Comparison of F1 Scores for Each Entity Type in ParsTwiNER

Entity Type	FaBERT	ParsBERT	AriaBERT	mBERT	XLM-R	Support
AerospaceManufacturer	0.7325	0.7127	0.7196	0.6269	0.638	1030
ORG	0.5809	0.5832	0.5348	0.5479	0.5325	18532
MusicalGRP	0.6282	0.6597	0.59	0.613	0.5954	4668
PrivateCorp	0.3822	0.4033	0.3851	0.2605	0.1749	148
CarManufacturer	0.6511	0.7031	0.6631	0.6291	0.6147	2085
PublicCorp	0.6109	0.6377	0.5819	0.5439	0.562	5926
SportsGRP	0.8159	0.8174	0.8012	0.8046	0.7949	6418
Medication/Vaccine	0.7067	0.6837	0.6342	0.6324	0.6582	4405
MedicalProcedure	0.6307	0.5965	0.5592	0.4904	0.5471	2132
AnatomicalStructure	0.6079	0.5827	0.5151	0.4824	0.4978	3940
Symptom	0.5656	0.5368	0.4671	0.4217	0.4109	821
Disease	0.646	0.6256	0.5737	0.5264	0.5652	3989
Artist	0.7384	0.7347	0.6936	0.7122	0.7155	51617
Politician	0.5786	0.6056	0.534	0.5213	0.5141	19760
Scientist	0.3328	0.3669	0.2952	0.2615	0.2625	3278
SportsManager	0.606	0.6232	0.5376	0.4332	0.4494	3009
Athlete	0.5796	0.5992	0.5356	0.5119	0.5357	12551
Cleric	0.5707	0.5535	0.4875	0.4627	0.4332	4526
OtherPER	0.4254	0.4225	0.3544	0.3647	0.3449	21127
Clothing	0.3912	0.3375	0.3293	0.2054	0.2716	239
Drink	0.5244	0.5683	0.5483	0.4646	0.5041	631
Food	0.6063	0.5971	0.574	0.4788	0.5591	3580
Vehicle	0.5388	0.5388	0.5171	0.4659	0.4952	2865
OtherPROD	0.5851	0.5843	0.5453	0.5109	0.5233	10897
ArtWork	0.0919	0.1085	0.1057	0.1077	0.0691	100
WrittenWork	0.5561	0.5541	0.5028	0.5006	0.5079	13530
VisualWork	0.7447	0.7463	0.7095	0.7445	0.7523	25054
Software	0.6448	0.6586	0.5991	0.5913	0.5911	8058
MusicalWork	0.5408	0.5714	0.5239	0.5492	0.545	6292
Facility	0.5673	0.5671	0.5283	0.5317	0.5347	11393
Station	0.7997	0.7863	0.7812	0.784	0.781	2532
HumanSettlement	0.7608	0.7676	0.7517	0.7658	0.7647	55741
OtherLOC	0.37	0.3348	0.3413	0.2965	0.2376	1241
Micro Average	0.6451	0.6517	0.6081	0.6108	0.6145	312115
Macro Average	0.5792	0.5809	0.54	0.5104	0.5147	312115
Weighted Average	0.6491	0.6531	0.6101	0.6111	0.6131	312115

Table 11: Comparison of F1 Scores for Each Entity Type in MultiCoNER v2