

PersianPunc: A Large-Scale Dataset and BERT-Based Approach for Persian Punctuation Restoration

Mohammad Javad Ranjbar Kalahroodi¹, Hesham Faili¹, Azadeh Shakery^{1,2}

¹University of Tehran, Tehran, Iran

²Institute for Research in Fundamental Sciences (IPM), Tehran, Iran
{mohammadjranjbar, hfaili, shakery}@ut.ac.ir

Abstract

Punctuation restoration is essential for improving the readability and downstream utility of automatic speech recognition (ASR) outputs, yet remains underexplored for Persian despite its importance. We introduce **PersianPunc**, a large-scale, high-quality dataset of 17 million samples for Persian punctuation restoration, constructed through systematic aggregation and filtering of existing textual resources. We formulate punctuation restoration as a token-level sequence labeling task and fine-tune ParsBERT to achieve strong performance. Through comparative evaluation, we demonstrate that while large language models can perform punctuation restoration, they suffer from critical limitations: over-correction tendencies that introduce undesired edits beyond punctuation insertion (particularly problematic for speech-to-text pipelines) and substantially higher computational requirements. Our lightweight BERT-based approach achieves a macro-averaged F1 score of 91.33% on our test set while maintaining efficiency suitable for real-time applications. We make our [dataset](#) and [model](#) publicly available to facilitate future research in Persian NLP and provide a scalable framework applicable to other morphologically rich, low-resource languages.¹

1 Introduction

Punctuation restoration represents an essential task in natural language processing, particularly for languages with limited computational resources. The absence of punctuation in raw text—whether from automatic speech recognition, informal digital communication, or historical documents—severely impacts the performance of downstream NLP tasks including machine translation, text summarization, and sentiment analysis.

¹Our resources are publicly available: [full dataset \(17M samples\)](#), [training subset](#), and [fine-tuned model](#).

Impact of Punctuation

Without punctuation: *bakhshesh lazem nist e'damesh konid*
Meaning: “No mercy needed, execute him” (Negative)

↓ Add comma

With punctuation: *bakhshesh, lazem nist e'damesh konid*
Meaning: “Forgiveness, no need to execute him” (Positive)

Without punctuation: *nah baba rast migi*
Tone: “No way, are you serious?” (Sarcastic)

↓ Add comma & period

With punctuation: *nah, baba rast migi.*
Tone: “No, dad, you’re right.” (Affirmative)

Figure 1: Persian punctuation restoration dramatically affects semantic interpretation. Minimal punctuation changes transform sentence meaning from negative to positive sentiment.

Despite the growing maturity of Persian NLP, punctuation restoration has received limited attention compared to other languages. The critical importance of punctuation in Persian is evidenced by dramatic semantic changes that occur with minimal punctuation modifications, as shown in Figure 1. Existing Persian studies have been constrained by small-scale datasets, domain-specific applications, or lack of publicly available models, highlighting the critical need for comprehensive approaches that can handle the full complexity of Persian text across diverse domains and writing styles.

This work addresses these challenges through a comprehensive approach to Persian punctuation restoration using fine-tuned BERT models and large-scale dataset curation. We present a dataset curation methodology that systematically aggregates multiple Persian text sources, resulting in a high-quality corpus for training robust punctuation restoration models. Our main contributions are:

- We present **PersianPunc**, a large-scale Persian punctuation restoration dataset containing 17 million filtered and deduplicated samples spanning diverse domains, sourced from six complementary corpora covering both formal and informal Persian text.
- We provide a systematic dataset curation framework including detailed preprocessing, quality filtering, and train/validation/test splits, with comprehensive analysis of punctuation distribution patterns in Persian.
- We achieve strong performance on Persian punctuation restoration with a fine-tuned ParsBERT model, demonstrating competitive results compared to large language models while requiring significantly lower computational resources and avoiding over-correction issues.

2 Related Work

Our research is situated at the intersection of punctuation restoration, Transformer-based NLP, and Persian text processing. This section reviews the evolution of methodologies for this task, starting from general approaches and progressively narrowing the focus to the specific challenges and prior work in the Persian language.

2.1 Punctuation Restoration as a Sequence Modeling Task

Historically, punctuation restoration was tackled with statistical methods, including n-gram language models (Beeferman et al., 1998; Gravano et al., 2009) and models incorporating prosodic features from speech to predict boundaries (Christensen et al., 2001; Kim and Woodland, 2003). These early systems laid the groundwork but were often limited by the scope of their handcrafted features and statistical models.

The advent of deep learning marked a significant shift. Recurrent Neural Networks (RNNs), particularly models using Bidirectional Long

Short-Term Memory (BiLSTM) units, became the standard, framing the problem as a sequence labeling task where each token is classified with a punctuation mark (or none) (Xu et al., 2016). This paradigm was often enhanced with Convolutional Neural Networks (CNNs) to capture local character-level features (Tündik and Szaszák, 2018; Zelasko et al., 2018), leading to substantial performance gains over classical methods. Our work follows this successful sequence labeling formulation, leveraging a more powerful neural architecture.

2.2 Transformer-Based Approaches for Punctuation Restoration

The introduction of the Transformer architecture (Vaswani et al., 2017) and pre-trained language models like BERT (Devlin et al., 2019) revolutionized NLP. For punctuation restoration, fine-tuning BERT-based models quickly became the state-of-the-art approach in high-resource languages, demonstrating superior performance in capturing long-range dependencies crucial for understanding sentence structure and punctuation placement (Courtland et al., 2020; Yi et al., 2020; Nagy et al., 2021). These models are typically lightweight and efficient, making them suitable for real-time applications like ASR post-processing.

More recently, Large Language Models (LLMs) have demonstrated impressive capabilities in a zero-shot or few-shot capacity for various text generation and correction tasks (Brown et al., 2020). However, their application to focused tasks like punctuation restoration comes with potential drawbacks, including high computational inference costs and a tendency for over-correction, where they may alter the source text beyond simply adding punctuation. A key part of our contribution is to rigorously evaluate these trade-offs against a fine-tuned, specialized model.

2.3 State of Persian Text Processing and Punctuation

Early work on Persian punctuation restoration was pioneering but limited in scale. Hosseini and Sameti (2017) introduced the first known corpus for this task, achieving an F1-score of 69.0% with a Conditional Random Field (CRF) model. While foundational, this work highlighted the need for larger and more diverse datasets and more powerful models.

More recently, Farokhshad et al. (2021) proposed Virapart, a multi-task text refinement framework for Persian that handles punctuation restoration, Zero-Width Non-Joiner (ZWNJ) recognition, and Ezafe construction. Using ParsBERT (Farahani et al., 2020), they achieved a strong F1-score of 92.13% for punctuation on the Bijankhan corpus (Bijankhan, 2004). Their work demonstrated the effectiveness of Transformer-based models for Persian text refinement. However, Virapart focuses on multiple text refinement tasks simultaneously, and was evaluated on a smaller, single-domain corpus.

Despite these advances, critical gaps remain. First, there is a lack of a large-scale, publicly available dataset specifically curated for punctuation restoration across diverse domains. Most existing efforts rely on smaller or general-purpose corpora like Bijankhan. Second, the capabilities and limitations of modern LLMs for Persian punctuation restoration have not been systematically studied, particularly regarding over-correction behavior.

3 Methodology

3.1 Dataset Construction

3.1.1 Data Sources and Collection Strategy

We construct a comprehensive Persian punctuation restoration dataset by systematically aggregating high-quality corpora spanning diverse domains and registers. Our multi-source approach addresses the linguistic diversity challenges of Persian NLP through careful curation. We selected source datasets through manual inspection, verifying that at least 100 random samples from each contained proper punctuation usage.

Our dataset combines sources across two primary categories:

Formal Academic Text: Bijankhan-Peykare Corpus (Bijankhan et al., 2011), Persian Medical QA (Kalahroodi et al., 2025), and Persian Wikipedia (MaralGPT, 2023) provide standardized punctuation patterns in formal contexts, covering literary, medical, and encyclopedic domains.

Contemporary Informal Text: Persian Telegram Channels (Shojaei, 2023), Farsi Stories (Pasban, 2023), and Blog Dataset V2 (Lab, 2023) capture modern conversational patterns and varied punctuation usage, representing social media, narrative fiction, and personal blogging styles.

3.1.2 Preprocessing and Quality Control

Normalization Pipeline All texts undergo systematic preprocessing to ensure consistency:

1. **Punctuation standardization:** English punctuation marks (comma, semicolon, question mark) are converted to their Persian equivalents (Persian comma `,`, Persian semicolon `؛`, Persian question mark `؟`).
2. **Character filtering:** Non-Persian characters are removed while preserving common Persian script variants and Arabic letters used in Persian text.
3. **Whitespace normalization:** Multiple spaces are collapsed, and leading/trailing whitespace is removed.

Sentence Segmentation and Filtering We first segment each source corpus into sentence-level units using end-of-sentence punctuation marks (period, exclamation mark, question mark). Each candidate sentence then undergoes multi-stage filtering:

- **Structural requirements:** Minimum length of 10 characters; at least two target punctuation marks from the set `{., , , !, ;, :}`; proper sentence termination with period, exclamation mark, or question mark.
- **Content filtering:** Removal of sentences containing URLs, email addresses, social media handles (`@` mentions), emojis, excessive special symbols (more than 20% non-alphabetic characters), or substantial mixed-language content (more than 30% non-Persian text).
- **Linguistic quality:** Pattern-based detection and removal of repetitive punctuation (e.g., `"..."`, `"!!!!"`), enumerative sequences (numbered lists, bullet points), and fragmented text (sentences with more than 50% single-character tokens).

Rationale for Filtering Criteria The requirement for at least two punctuation marks ensures that samples present meaningful punctuation restoration challenges beyond simple sentence termination. While this filtering criterion does exclude simple sentences (which are underrepresented in formal Persian writing), it ensures the

dataset focuses on the core challenge of internal sentence punctuation, which is critical for ASR applications where sentence boundaries are often detected separately. We acknowledge this as a dataset characteristic rather than a limitation, as it creates a focused benchmark for comma, colon, and question mark insertion—the most challenging and impactful aspects of Persian punctuation restoration. Future work could address simple sentence coverage through stratified sampling or separate evaluation sets.

3.1.3 Deduplication and Dataset Splitting

To ensure dataset quality and prevent data leakage, we perform exact deduplication across all source corpora. Due to the large dataset size (initial pool of over 20 million samples), we implement an efficient SHA-256 hash-based deduplication strategy with whitespace normalization. Each sentence is normalized (lowercased, whitespace-collapsed) before hashing to detect duplicates that differ only in formatting.

After deduplication, our final dataset contains 17,102,014 unique samples. For model training and evaluation, we randomly sample a 1M subset stratified by source corpus. This subset is split into training (989,000 samples), validation (10,000 samples), and test (1,000 samples) sets. The sampling strategy maintains the source distribution proportions to ensure representativeness across domains.

3.2 Dataset Statistics and Punctuation Analysis

We conducted a comprehensive punctuation analysis on the complete dataset of 17,102,014 samples to understand the characteristics of Persian punctuation usage in our corpus.

3.2.1 Punctuation Distribution

Table 1 presents the distribution of punctuation marks across the entire dataset. All samples contain at least one punctuation mark (by design), with an average of 2.51 punctuation marks per sentence.

The distribution reflects typical Persian text characteristics, where commas are heavily used for clause separation and complex sentence structures.

3.3 Punctuation Restoration Model and Training Setup

We formulate punctuation restoration as a token-level sequence labeling problem. Given an input

Table 1: Distribution of punctuation marks in the complete dataset (17M samples).

Mark	Total Count	% of Total
Persian comma (.)	21,291,632	50.13%
Period (.)	15,076,946	35.50%
Colon (:)	4,228,554	9.96%
Exclamation (!)	1,209,227	2.85%
Persian question (؟)	665,841	1.57%
Total	42,472,200	100.00%

sequence of tokens without punctuation, the model predicts a punctuation label for each token position. We define five classes: EMPTY (no punctuation), COMMA (.), QUESTION (؟), PERIOD (.), and COLON (:). Note that we focus on the four most common and semantically important punctuation marks, excluding exclamation marks and semicolons which are less frequent and often interchangeable with periods and commas in Persian.

Model Architecture Our model architecture consists of a pre-trained ParsBERT encoder (Farahani et al., 2020) followed by a linear classification layer with dropout regularization. ParsBERT is a monolingual Persian BERT model pre-trained on a large Persian corpus, making it well-suited for Persian NLP tasks.

Given an input sentence with punctuation removed, we:

1. Tokenize using ParsBERT’s WordPiece tokenizer
2. Pass tokens through ParsBERT to obtain contextualized embeddings
3. Apply dropout ($p=0.1$) for regularization
4. Project embeddings to 5-dimensional class logits via a linear layer
5. Assign the predicted punctuation class to each token position

For subword tokens generated by WordPiece tokenization, we assign punctuation labels only to the first subword of each word, ignoring continuation subwords during both training and evaluation. This aligns the token-level predictions with word-level punctuation placement.

Training Configuration We train on the 1M sample subset described in Section 3.1.3. The model is optimized using AdamW with a learning rate of 2×10^{-5} , weight decay of 0.01, and trained for 3 epochs. We use a batch size of 85 with gradient accumulation over 8 steps (effective batch size

of 680). The loss function is cross-entropy computed over all token positions.

Evaluation Metrics We employ standard sequence labeling metrics:

- **Per-class metrics:** Precision, recall, and F1-score for each punctuation class (COMMA, PERIOD, QUESTION, COLON)
- **Macro-averaged F1:** Arithmetic mean of per-class F1-scores, giving equal weight to each punctuation type regardless of frequency
- **Micro-averaged F1:** F1-score computed from the sum of per-class true positives, false positives, and false negatives, effectively weighting classes by their frequency
- **Full Sentence Match (FSM) Rate:** Percentage of test sentences where the predicted punctuation sequence exactly matches the gold standard. This metric is particularly important for evaluating LLMs, as it captures whether the model made any edits beyond punctuation insertion (over-correction).

Throughout this paper, unless otherwise specified, “F1-score” refers to the macro-averaged F1-score, which provides an overall measure of punctuation restoration accuracy giving equal weight to each punctuation type regardless of frequency.

4 Results and Analysis

4.1 Overall Performance

Our fine-tuned ParsBERT model achieves a macro-averaged F1-score of 91.33% and a micro-averaged F1-score of 97.28%. The macro-averaged score is lower due to the class imbalance (periods and commas dominate the dataset), but performance remains strong across all punctuation types as shown in Table 6 in the appendix.

4.2 Comparison with Prior Work

It is important to note that the CRF and ViraPart results shown in Table 2 are evaluated on different datasets (the Hosseini et al. corpus and Bijankhan corpus, respectively), and therefore cannot be directly compared to our results. We include these numbers for reference to situate our work within the Persian punctuation restoration literature, but we make no claims of superiority over these methods without evaluation on the same test set.

The substantial improvement over the CRF baseline (69.00% vs 91.33%) likely reflects both the advancement in modeling approaches (Transformer-based vs. CRF) and potential differences in dataset difficulty. The ViraPart score (92.13%) is more competitive, though direct comparison remains inappropriate due to the different evaluation sets.

4.3 Comparison with Large Language Models

We evaluated two variants of GPT-4o on our test set using the zero-shot prompt shown in Appendix 2. The prompt explicitly instructs the model to only add punctuation without modifying the source text.

Our ParsBERT model achieves 91.33% macro F1, outperforming both GPT-4o (85.96%) and GPT-4o-mini (79.54%) on the same test set. More importantly, the FSM Rate reveals a critical limitation of LLM-based approaches: GPT-4o achieves only 50.10% exact matches, while our model achieves 61.80%.

Analysis of the mismatches reveals that GPT-4o exhibits over-correction in approximately 5% of samples, making undesired edits such as:

- Removing words deemed unnecessary
- Replacing informal words with formal equivalents
- Correcting perceived spelling or grammatical errors

Notably, we observed no cases of word additions, only deletions and substitutions. This over-correction behavior is particularly problematic for ASR post-processing pipelines, where the source text (transcribed speech) should be preserved verbatim with only punctuation added.

Additionally, GPT-4o requires substantially higher computational resources for inference compared to our lightweight ParsBERT model, making it less suitable for real-time applications or deployment in resource-constrained environments.

4.4 Analysis of Model Performance

Table 6 (Appendix) provides detailed per-class performance metrics. The model performs exceptionally well on periods (F1: 98.71%), which is expected given their high frequency and relatively consistent usage patterns. Performance on other punctuation types remains strong: colons (90.45%), question marks (88.89%), and commas (80.03%).

Model	Test Set	Macro F1 (%)	FSM (%)
CRF (Hosseini and Sameti, 2017)	Hosseini et al. corpus	69.00	—
ViraPart (Farokhshad et al., 2021)	Bijankhan corpus	92.13	—
GPT-4o-mini	Our test set	79.54	38.01
GPT-4o (OpenAI, 2023)	Our test set	85.96	50.10
Our Model (ParsBERT)	Our test set	91.33	61.80

Table 2: Comparison of punctuation restoration performance across models. Note that CRF and ViraPart results are on different test sets and are not directly comparable. GPT-4o and our model are evaluated on the same test set from PersianPunc.

The lower performance on commas reflects their more nuanced usage in Persian, where comma placement can be somewhat flexible and context-dependent, leading to greater ambiguity in the gold standard annotations themselves.

5 Conclusion and Future Work

This work presents PersianPunc, a large-scale dataset of 17 million samples for Persian punctuation restoration, constructed through systematic aggregation and quality filtering of diverse Persian text sources. We demonstrate that a fine-tuned ParsBERT model achieves strong performance (91.33% macro F1) while avoiding the over-correction issues and computational overhead of large language models.

Our primary contribution is the dataset itself, which addresses a critical gap in Persian NLP resources. The curation methodology, including detailed preprocessing pipelines, quality filtering criteria, and comprehensive punctuation analysis, provides a framework applicable to other low-resource languages.

Future work should explore several directions. The development of domain-specific models for Persian literature, news, and social media text could address the variation in punctuation usage across different domains. Additionally, incorporating prosodic information from Persian speech could improve punctuation restoration for speech-to-text applications. Furthermore, extending the model to jointly handle punctuation restoration and Zero-Width Non-Joiner (ZWNJ) insertion would address the broader text normalization challenges specific to Persian writing systems.

6 Limitations

This work has several limitations that should be acknowledged. First, the dataset creation pro-

cess relies on existing Persian texts, which may contain punctuation errors or inconsistencies that could propagate to the trained model. Second, the model’s performance is optimized for contemporary Persian writing styles and may not generalize well to historical or highly specialized Persian texts. Third, our evaluation is limited to 1,000 test sentences due to resource constraints, including expensive API costs for commercial LLM evaluation and lack of GPU access for extensive experimentation. More extensive evaluation with larger test sets, multiple training runs, and statistical significance testing would strengthen our findings but was not feasible given these constraints.

References

- D Beeferman, A Berger, and J Lafferty. 1998. Cyberpunc: a lightweight punctuation annotation system for speech. In *ICASSP*, pages 689–692.
- M Bijankhan. 2004. The role of the corpus in writing a grammar: An introduction to a software. *Iranian Journal of Linguistics*, 19(2):48–67.
- Mahmood Bijankhan, Javad Sheykhzadegan, Mohammad Bahrani, and Masood Ghayoomi. 2011. [Lessons from building a persian written corpus: Peykare](#). *Language Resources and Evaluation*, 45(2):143–164.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *Advances in neural information processing systems*, volume 33, pages 1877–1901.
- H Christensen, Y Gotoh, and S Renals. 2001. Punctuation annotation using statistical prosody mod-

- els. In *Proc Isca Workshop on Prosody in Speech Recognition and Understanding*.
- M Courtland, A Faulkner, and G McElvain. 2020. Efficient automatic punctuation restoration using bidirectional transformers with robust inference. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 272–279.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [Bert: Pre-training of deep bidirectional transformers for language understanding](#). *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.
- Mehrdad Farahani, Mohammad Gharachorloo, Marzieh Farahani, and M Manthouri. 2020. Parsbert: Transformer-based model for persian language understanding. *arXiv preprint arXiv:2005.12515*.
- Narges Farokhshad, Milad Molazadeh, Saman Jamalabasi, Hamed Babaei Giglou, and Saeed Bibak. 2021. Virapart: A text refinement framework for automatic speech recognition and natural language processing tasks in persian. In *arXiv preprint arXiv:2110.09086v3*.
- A Gravano, M Jansche, and M Bacchiani. 2009. Restoring punctuation and capitalization in transcribed speech. In *ICASSP*, pages 4741–4744.
- Mohammadsaleh Hosseini and Hossein Sameti. 2017. [Creating a corpus for automatic punctuation prediction in persian texts](#). In *Proceedings of the 2017 Iranian Conference on Electrical Engineering (ICEE)*, Tehran, Iran.
- Mohammad Javad Ranjbar Kalahroodi, Amirhossein Sheikholeslami, Sepehr Karimi, Sepideh Ranjbar Kalahroodi, Hesham Faily, and Azadeh Shakery. 2025. [Persianmedqa: Evaluating large language models on a persian-english bilingual medical question answering benchmark](#). *Preprint*, arXiv:2506.00250.
- J Kim and P Woodland. 2003. A combined punctuation generation and speech recognition system and its performance enhancement using prosody. *Speech Communication*, 41:563–577.
- RohanAI Lab. 2023. Persian blog dataset version 2 for text analysis. https://huggingface.co/datasets/RohanAiLab/persian_blog_V2.
- MaralGPT. 2023. Persian wikipedia dataset for large language model training. <https://huggingface.co/datasets/MaralGPT/persian-wikipedia>.
- Attila Nagy, Bence Bial, and Judit Ács. 2021. [Automatic punctuation restoration with bert models](#). *arXiv preprint arXiv:2101.07343*.
- OpenAI. 2023. [Gpt-4 technical report](#). *arXiv preprint arXiv:2303.08774*.
- Sina Pasban. 2023. Farsi tiny stories: A persian dataset for language model training. <https://huggingface.co/datasets/sinap/FarsiTinyStories>.
- Mohammad Shojaei. 2023. Persian telegram channels dataset for nlp research. <https://huggingface.co/datasets/mshojaei77/PersianTelegramChannels>.
- Márk Á Tündik and György Szaszák. 2018. Joint word- and character-level embedding cnn-rnn models for punctuation restoration. In *2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pages 135–140.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.
- Kaituo Xu, Lei Xie, and Kaisheng Yao. 2016. Investigating lstm for punctuation prediction. In *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 1–5.
- Jingsi Yi, Jianhua Tao, Zheng Tian, Ye Bai, and Cunli Fan. 2020. Focal loss for punctuation prediction. In *Proc. Interspeech*, pages 721–725.
- Piotr Zelasko, Piotr Szymanski, Jan Mizgajski, Adrian Szymczak, Yishay Carmiel, and Najim Dehak. 2018. Punctuation prediction model for conversational speech. In *Proc. Interspeech*, pages 3603–3607.

A Punctuation Analysis Details

In this appendix, we provide comprehensive analysis of punctuation patterns and model performance.

A.1 Punctuation Co-occurrences

Analysis of punctuation co-occurrence reveals common patterns in Persian writing. Table 3 shows the most frequent punctuation pairs appearing together in the same sentence.

Table 3: Most frequent punctuation co-occurrences in the dataset.

Punctuation Pair	% of Sentences
Period + Persian comma	79.43%
Period + Colon	16.91%
Colon + Persian comma	10.94%
Exclamation + Persian comma	4.34%

The combination of period and Persian comma appears in nearly 80% of sentences, indicating that most sentences contain multiple clauses separated by commas before the final period.

A.2 Sentence-Level Punctuation Coverage

Table 4 shows the percentage of sentences containing each punctuation mark (counted once per sentence regardless of frequency).

Table 4: Percentage of sentences containing each punctuation mark.

Punctuation	Coverage
Period (.)	15,076,946 (88.94%)
Persian comma (.)	14,585,086 (86.04%)
Colon (:)	4,036,797 (23.81%)
Persian question (؟)	665,841 (3.93%)

A.3 Distribution of Punctuation Counts per Sentence

Table 5 presents the distribution of the number of punctuation marks per sentence. The majority of sentences (68.37%) contain exactly 2 punctuation marks, which is a direct consequence of our filtering criterion requiring at least 2 marks per sentence combined with the natural distribution in source texts.

Table 5: Distribution of punctuation counts per sentence.

# Punctuations	# Sentences	Percentage
2	11,589,324	68.37%
3	3,420,146	20.18%
4	1,034,579	6.10%
5	362,609	2.14%
6+	511,518	3.02%
Total	17,102,014	100.00%

A.4 Punctuation-Specific Performance

Table 6 presents a detailed analysis of per-class performance. The macro-averaged F1-score of 91.33% demonstrates strong overall performance across all punctuation classes.

Table 6: Per-class performance metrics for punctuation restoration on the test set (1,000 sentences).

Punctuation	Precision	Recall	F1-Score
Persian Comma (.)	0.8408	0.7635	0.8003
Period (.)	0.9855	0.9886	0.9871
Question (؟)	0.8750	0.9032	0.8889
Colon (:)	0.9137	0.8955	0.9045
Macro Average	0.9038	0.8877	0.9202
Micro Average	0.9729	0.9727	0.9728

B LLM Evaluation Prompt

We used the prompt shown in Figure 2 to evaluate GPT-4o and GPT-4o-mini. The temperature was set to 0, and maximum tokens were set to 2048 to accommodate longer outputs. Prompts were issued in English, as we found LLMs demonstrate better instruction-following in English compared to Persian.

Evaluation Prompt for LLMs

Role: You are a punctuation restoration system for Persian text.

Task: Add appropriate punctuation marks to the given Persian text.

Rules:

- Do NOT fix, correct, or modify ANY words in the text.
- Do NOT change the order of words.
- Do NOT add or remove any words.
- ONLY add punctuation marks where appropriate.
- Use these punctuation marks: . (period), , (Persian comma), ؟ (Persian question mark), : (colon)
- Return the result as a JSON object with a single key "text" containing the punctuated text.

Input text (without punctuation): *{text}*

Output format:

```
{"text": "your punctuated text here"}
```

Important: Keep ALL words EXACTLY as they are in the input. Do NOT fix spelling, grammar, or anything else. ONLY add punctuation.

Figure 2: Prompt used for zero-shot evaluation of GPT-4o and GPT-4o-mini on Persian punctuation restoration. The system is explicitly instructed to only add punctuation marks without altering the original text in any way.