Matina: A Large-Scale 73B Token Persian Text Corpus

Sara Bourbour Hosseinbeigi
Tarbiat Modares University
s.bourbour@modares.ac.ir

Fatemeh Taherinezhad
University of Tehran
ftaherinezhad@ut.ac.ir

Heshaam Faili
University of Tehran
hfaili@ut.ac.ir

Hamed Baghbani
University of Tehran
baghbani.hamed@ut.ac.ir

Fatemeh Nadi
University of Tehran
fatemehnadi@ut.ac.ir

Mostafa Amiri
University of Tehran
mostafa.amiri@ut.ac.ir

Abstract

Text corpora are essential for training models used in tasks like summarization, translation, and large language models (LLMs). While various efforts have been made to collect monolingual and multilingual datasets in many languages, Persian has often been underrepresented due to limited resources for data collection and preprocessing. Existing Persian datasets are typically small and lack content diversity, consisting mainly of weblogs and news articles. This shortage of high-quality, varied data has slowed the development of NLP models and open-source LLMs for Persian. Since model performance depends heavily on the quality of training data, we address this gap by introducing the Matina corpus, a new Persian dataset of 72.9B tokens, carefully preprocessed and deduplicated to ensure high data quality. We further assess its effectiveness by training and evaluating transformer-based models on key NLP tasks. Both the dataset and preprocessing codes are publicly available[1], enabling researchers to build on and improve this resource for future Persian NLP advancements.

1 Introduction

Since the introduction of the transformer architecture (Vaswani, 2017), natural language processing (NLP) has advanced rapidly, transforming many language-related tasks. Transformer-based models, like BERT (Devlin, 2018) and GPT-2 (Radford, 2018), initially focused on tasks like sentiment analysis, translation, and summarization. However, with the development of large-scale language models (LLMs), such as GPT-3 (Brown, 2020) and later models (Touvron et al., 2023; Le Scao et al., 2023; Bai et al., 2023; Yang et al., 2024), the research shifted towards more complex tasks, including generalization, creative problem-solving, and critical thinking.

The performance of these models, in both basic and advanced tasks, isn't just about model size or computational power—it's also heavily influenced by the quality and amount of training data. As a result, a lot of effort has gone into large-scale data collection and preprocessing (Gao et al., 2020; Laurençon et al., 2022; Penedo et al., 2023) to improve model capabilities and generalization.

While English dominates NLP research, there has been a growing effort to curate multilingual datasets (Wenzek et al., 2019; Laurençon et al., 2022; Nguyen et al., 2023; Kudugunta et al., 2024) and develop models capable of understanding multiple languages (Le Scao et al., 2023; Touvron et al., 2023; Yang et al., 2024).

Despite Persian being widely spoken, it remains underrepresented in NLP research. Although both conventional models and LLMs can process Persian, their performance is often suboptimal, mainly because of the limited availability and poor quality of existing data. Persian text data is predominantly sourced from news websites and blogs, which often lack formal or factual content. Moreover, no standardized preprocessing pipeline exists to ensure the high quality of Persian datasets at the same level as those available for other languages.

To address this gap, we introduce the Matina Corpus, a 72.9 billion token Persian dataset designed for training language models. Unlike other Persian datasets (Targoman, 2022; Sabeti et al., 2018), the Matina Corpus has undergone a rigorous and well-designed preprocessing pipeline and a comprehensive deduplication process to ensure its high quality. The dataset includes not only publicly available Persian datasets but also introduces newly collected sources to ensure greater diversity and the inclusion of factual information. The diverse sources in the dataset make it suitable both for training large language models and for a variety of downstream tasks that require clean, high-quality Persian data.

[1] https://github.com/FTaheriN/Matina-Text-Preprocessing

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 9143–9157

Component	Number of tokens	Mean document length
Books	2,842,128,225	162,648.9
Papers	3,547,046,981	10,620.5
Social Media	2,143,415,349	351.6
Web Crawled	14,782,414,716	749.7
CulturaX FA	20,469,778,795	1,124.8
MADLAD-400 FA	29,131,569,264	1,352.96
Matina	72,916,353,330	1,106.5

Table 1: Overview of components in Matina Corpus. Tokens are counted by the Llama 3.1 (Dubey et al., 2024) tokenizer.

The Matina Corpus includes Persian sections from Madlad (Kudugunta et al., 2024), CulturaX (Nguyen et al., 2023), and the most recent Persian Wikipedia update. Each data source was processed differently, based on heuristics derived from careful evaluation and observation of the content. To ensure quality and avoid redundancy, deduplication was applied to related chunks of documents rather than across the entire dataset at once. The final corpus comprises a total of 72.9 B tokens, with an average document length of 1,106.5 across different sources (as summarized in Table 1), illustrating both the breadth and depth of the dataset.

The Matina Corpus is designed to enhance Persian NLP by supporting both the pretraining of large language models (LLMs) and the development of smaller models based on transformers and other architectures. It enables various NLP tasks, including text classification, machine translation, and sentiment analysis. To evaluate its impact, we continued the pretraining of XML-RoBERTa (Conneau, 2019a) on Matina and assessed its performance on sentiment analysis, text emotion detection, and named entity recognition, observing notable improvements over models trained on existing Persian datasets.

Furthermore, integrating this high-quality dataset into multilingual models enhances their Persian language comprehension, helping bridge the resource gap. To measure this effect, we used portions of the corpus to continue pretraining LLaMA 3.1 8B, achieving significant gains in Persian text understanding.

The rest of this paper is structured as follows: We begin by providing an overview of existing large corpora, along with the preprocessing pipelines applied to them, covering English, multilingual, and Persian datasets. Afterward, we introduce our corpus, dividing it into three distinct sections based on content, and offer details on the preprocessing steps we applied. We then assess the dataset's effectiveness through model training and evaluation. Finally, we analyze the dataset, discuss its limitations, and conclude with a summary of our dataset.

2 Related Work

The scope of our dataset encompasses two key dimensions: (1) the preprocessing steps involved in creating large-scale corpora and (2) the development of extensive text corpora in Persian. Accordingly, we divide this section into two parts. First, we review notable large-scale corpora available in languages other than Persian, along with the preprocessing techniques applied to these datasets. Then, we examine and analyze the current state of publicly available Persian corpora.

2.1 Large-Scale Public Corpora

Since the early stages of NLP development, there have been efforts to compile large-scale datasets for training models in various downstream tasks, such as sentiment analysis, summarization, and text classification, among others (Glockner et al., 2018; Narayan et al., 2018; Wang et al., 2019). With the advent of deep learning models, these efforts have escalated in scope, culminating in the large-scale data collection necessary for training large language models (LLMs). One of the earliest and most significant contributions to the development of large text corpora is Common Crawl (Crawl, 2008).

Common Crawl (Crawl, 2008) is a vast multilingual web corpus that continuously archives webpage data from the Internet. However, Common Crawl contains substantial amounts of extraneous content, including advertisements, navigation bars, and inappropriate materials such as pornography, violence, spam, and sensitive personal information.

In response to these issues, datasets like OSCAR (Suárez et al., 2019), C4 (Raffel et al., 2020), mC4 (Xue, 2020), The Pile (Gao et al., 2020) RefinedWeb (Penedo et al., 2023), and FineWeb (Penedo et al., 2024) have been created to provide cleaner and more refined versions of the Common Crawl data.

Suárez et al. (2019) took a parallel method to fastText (Athiwaratkun et al., 2018) when preprocessing Common Crawl for better data quality. A linear classifier was used to categorize the WET files for language, followed by a filter for erroneous UTF-8 characters and a hashing approach to remove duplicates. This approach produced a 6.3TB dataset covering 160 languages. Similar pipelines were used to build datasets such as CC-100 (Conneau, 2019b) and RedPajama (Computer, 2023).

Likewise, C4 (Raffel et al., 2020) was constructed from Common Crawl data to train the T5 model. The langdetect[2] tool was employed to filter only English pages. Pages containing inappropriate content, specific keywords, curly brackets (identified as code), or a limited number of lines were removed. Subsequently, a set of heuristics was applied at the line level, including checks for terminal punctuation, JavaScript keywords, and boilerplate text. The documents were then deduplicated using a three-sentence span. Building upon C4 (Raffel et al., 2020), mC4 (Xue, 2020) expanded the dataset to 107 languages. Cld[3] was used for language classification, and documents with language confidence below 70% were discarded. As in C4 (Raffel et al., 2020), deduplication was performed at the final stage.

Due to the limited factual and academic content in previous datasets, The Pile (Gao et al., 2020) introduced 21 additional sources, including books, academic papers, code, and subtitles, alongside Common Crawl data (Pile CC). Each data source was processed using specific heuristics tailored to its structure, and the sources were unified into an English-only dataset of 825 GiB. Similarly, MassiveText (Muennighoff et al., 2022) was created to train the Gopher model, drawing from six sources: massiveWeb, books, C4, news, GitHub, and Wikipedia. The web data was filtered based on various criteria, including non-English content, fewer than two English stopwords, excessive bullet points, unsuitable word count or length, and pages with repeated words or phrases. Deduplication was performed using MinHash (Broder, 1997) with Jaccard similarity, producing a multilingual dataset containing 2.53 billion documents.

ROOTS (Laurençon et al., 2022) is a multilingual corpus that includes 46 natural and 13 programming languages. Although the 1.6TB collection comprises primarily of web-based information, many websites were created through crowdsourcing. Pages were filtered using heuristics and thresholds, with low-quality documents deleted using a pretrained tokenizer. Personal information such as email addresses, phone numbers, and IP addresses were eliminated with regular expressions. To assure data quality, the crowd workers selected language-specific preprocessing methods.

RefinedWeb (Penedo et al., 2023) used a similar preprocessing pipeline to MassiveText, with additional heuristics for document filtering. Starting with web-based data from multiple Common Crawl dumps, English documents were first identified using fastText (Athiwaratkun et al., 2018) and then filtered at both the document and line levels. More strict filtering was applied to remove sensitive and adult content. Deduplication was performed using both fuzzy methods and exact substring matching. RedPajamas v2 (Computer, 2023) was created using 84 Common Crawl dumps and the CC-Net (Wenzek et al., 2019) preprocessing pipeline, with fuzzy and exact-matching deduplication. This dataset spans five languages and contains 100 billion documents. Building on their earlier dataset, Huggingface introduced FineWeb (Penedo et al., 2024) based on 95 Common Crawl snapshots. After following a similar preprocessing procedure to RefinedWeb (Penedo et al., 2023), additional heuristics and a different deduplication method, derived from extensive ablation studies, were applied. The final processed dataset is 96.4TB in size.

2.2 Persian Text Corpora

The rapid development of natural language processing (NLP) has necessitated the creation of diverse, large-scale text corpora across various languages. For Persian, also known as Farsi, the availability of robust datasets is crucial for enhancing language modeling capabilities. However, a significant gap persists in terms of corpora that are sufficiently diverse and preprocessed for effective use in training LLMs. Many existing Persian datasets predominantly feature news content, which does not ade-

[2] https://pypi.org/project/langdetect/
[3] https://github.com/google/cld3

quately cover the full spectrum of language use. Despite these limitations, Persian remains a language with rich literary and cultural resources, suggesting a substantial potential for corpus development.

Several Persian corpora, including the Persian Wikipedia Corpus[4], MirasText[5], hmBlogs (Khansari and Shamsfard, 2021), Naab (Sabouri et al., 2022), Targoman (Targoman, 2022), have significantly enriched the pool of publicly available Persian data. The Persian Wikipedia Corpus, with over one million articles, serves as a foundational resource, though its content is mainly formal and factual. MirasText, covering 2.8 million articles from more than 250 news websites, and Naab (Sabouri et al., 2022), containing around 15 billion tokens, both contribute vast data but are largely news-centric, which limits content diversity. In contrast, Targoman (Targoman, 2022) expands the scope by incorporating 65 million documents across weblogs, forums, literature, and educational content, although issues with licensing and accessibility hinder its public use. Additionally, hmBlogs (Khansari and Shamsfard, 2021) offers a valuable glimpse into colloquial language with 20 million blog posts spanning 15 years, though it requires extensive preprocessing to ensure its consistency and applicability. Additionally, Ganjoor[6] introduces classical Persian poetry from 12 poets, enhancing the stylistic and lexical range of the corpus and providing unique linguistic depth.

Parallel corpora, including TEP: Tehran English-Persian parallel corpus (Tiedemann, 2012), MIZAN (Kashefi, 2020), and the Bible Corpus[7], further extend the utility of Persian datasets by enabling translation tasks and bilingual language modeling. MIZAN (Kashefi, 2020), containing one million sentence pairings between Persian and English, allows cross-linguistic studies and machine translation. However, the breadth of such corpora is frequently limited.

Standardized preprocessing techniques are required to improve Persian language modeling by filtering non-Farsi words, unifying Arabic and Farsi characters, and removing unnecessary content. These steps are crucial for creating a high-quality, clean corpus, as data quality directly impacts model performance in large language models (LLMs). While some datasets, such as Naab (Sabouri et al., 2022) and hmBlogs (Khansari and Shamsfard, 2021), offer preprocessed versions, this is still the exception rather than the norm in Persian corpus development.

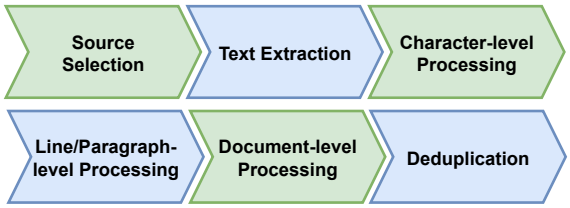

Figure 1: The overall stages of processing pipeline of Matina Corpus.

3 Matina Corpus

The Matina corpus is built from a variety of data sources, each of which is processed based on its specific content characteristics. Although these sources are grouped into three main categories, the overall preprocessing pipeline remains consistent, as depicted in Figure 1, with variations primarily in hyperparameters. Certain sources, however, demand additional cleaning steps, which are detailed in their respective sections.

Figure 2 visualizes the distribution of token counts across documents from each source, using a box plot to illustrate the variance in document length. These three categories—web-based crawled data, crawled books and papers, and social media—form the core of our dataset, each with distinct preprocessing requirements. In this section, we describe the data collection process, the preprocessing techniques applied, and the rationale behind the decisions made throughout these steps.

3.1 Web-based Crawled Data

Web crawling is a common and efficient method for collecting data in any language. Websites offer a vast range of valuable information and, given their structured nature and wide availability, can largely be crawled automatically. As a result, web data is frequently used as the primary source for constructing large-scale text datasets. However, while the bulk collection of web data is straightforward, extracting meaningful content from irrelevant elements such as metadata, advertisements, and embedded links remains challenging. Web pages often

[4] https://github.com/Text-Mining/Persian-Wikipedia-Corpus
[5] https://github.com/miras-tech/MirasText
[6] https://github.com/ganjoor
[7] https://github.com/christos-c/bible-corpus

contain spam-like elements, which complicates the cleaning process and increases the likelihood of errors.

Most web-based datasets begin with basic steps such as text extraction and language detection, often followed by optional URL filtering to exclude content deemed inappropriate or irrelevant. Further preprocessing steps are applied, followed by deduplication to ensure data quality and minimize redundancy. We adopt a similar approach in preprocessing the web data collected for the Matina corpus.

Matina's web-based data is divided into two parts: data crawled by our team and data taken from two public databases using the Common Crawl (Crawl, 2008) dataset. This dual-source strategy uses both proprietary and publically available data to increase the corpus's breadth and diversity.

In any language, certain domains are recognized for their reliability and high-quality information. We identified such domains in Persian and crawled them to extract relevant textual content. This step helped minimize the inclusion of irrelevant elements such as advertisements, tags, or comments. Text extracted from headings and paragraphs was merged to form unified documents, with additional informative fields (e.g., summaries or subheadings) incorporated as metadata, if available. Because these domains were manually selected, language detection and URL filtering were unnecessary. We also ensured that the selected URLs did not contain harmful, sensitive, or adult content.

For the public datasets, Madlad-400 (Kudugunta et al., 2024) and CulturaX (Nguyen et al., 2023), the initial preprocessing steps—such as language detection, text extraction, and URL filtering—had already been completed by the dataset providers. These datasets also included filters for toxic or harmful content, which allowed us to directly proceed to the next stages of preprocessing. While both datasets applied generic filters—such as language mismatch detection, character ratio checks, and word/sentence length thresholds, these filters were not language-specific. Therefore, we processed data from these sources similarly to the web data we crawled ourselves. After applying the processing on data sourced from web and the public datasets, there remained 64.3B tokens with an average document length of 1,141.8 tokens.

After inspecting samples from various domains, we defined heuristic functions to modify documents and remove those deemed irrelevant. These heuristics were inspired by preprocessing pipelines adopted in BLOOM (Le Scao et al., 2023), MassiveText (Muennighoff et al., 2022), and RefineWeb (Penedo et al., 2023), but we tailored them to the specific characteristics of our data and added multiple other processing functions.

Our preprocessing pipeline for web-based data encompasses three primary stages: character-level processing, line and paragraph-level processing, and document-level processing. Each stage employs a series of targeted operations to enhance data quality, ensure linguistic consistency, and eliminate redundancies. Appendix A provides a full explanation of each step in the preprocessing and deduplication procedures.

Character-level processing involves normalizing Persian characters, mapping symbols and numbers to their Persian equivalents, limiting the occurrence of repeated characters, standardizing newline characters, and removing non-standard Unicode symbols. This stage ensures that the text adheres to consistent encoding standards and minimizes the presence of corrupted or irrelevant characters.

Line and paragraph-level processing focuses on the structural integrity of the text by removing HTML and JavaScript tags, handling custom structures specific to certain domains, filtering out lines with excessive special characters, and eliminating short or incomplete lines that do not contribute meaningful content.

Document-level processing entails a comprehensive evaluation of each document's relevance and quality. Documents are discarded based on criteria such as insufficient length, predominance of non-Persian content, excessive repetition of words, high proportion of short lines, and the presence of out-of-vocabulary (OOV) words. These filters ensure that only high-quality, relevant, and linguistically coherent documents are retained in the corpus.

After cleaning the documents, we apply a deduplication step to mitigate data redundancy, a crucial aspect of the preprocessing pipeline highlighted in several studies (Gao et al., 2020; Penedo et al., 2023; Le Scao et al., 2023). Utilizing the MinHash algorithm (Broder, 1997), we efficiently identify and eliminate both exact and near-duplicate documents, thereby enhancing the corpus's uniqueness.

For two manually inspected domains, Virgool[8]

[8] https://virgool.io/

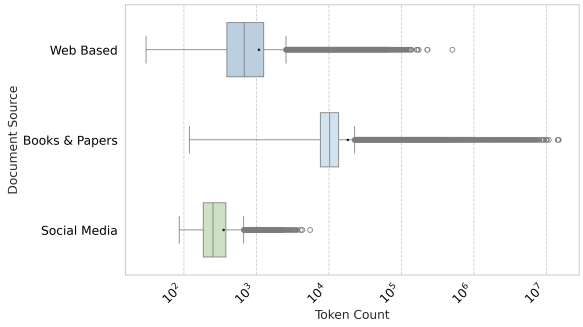

Figure 2: Distribution of document length by source in the Matina Corpus. Length is determined by the log of the number of tokens using Llama3.1 (Dubey et al., 2024) tokenizer.

and WikiShia[9], we adopted a tailored processing approach to account for domain-specific characteristics. Virgool's diverse blog posts required relaxed filtering criteria to preserve technical content, while WikiShia's recursive linking and bilingual content deemed for specialized deduplication and language handling techniques to maintain content integrity and cultural relevance.

3.2 Crawled Books and Papers

Data collected from the web alone does not provide sufficient factual or literary content. To enrich our dataset, we also sourced publicly accessible books and academic papers from websites and social media channels. As demonstrated in Figure 2, the box plot of document length distribution clearly shows that books and papers contain significantly longer texts compared to web and social media content, making them more informative and comprehensive. This length, along with the depth of the content, further justifies the inclusion of these sources in our corpus.

Since most of these sources provide data in PDF format, additional steps were required to convert PDFs into usable text. However, the limited accuracy of Persian OCR systems introduces challenges, particularly when processing PDFs that contain scanned images.

We divided the data from books and papers into two groups, each requiring different processing steps based on the nature of the data: Text-based PDFs and Image-based PDFs (OCR). Just like the data from web, the processing of books and papers involved a combination of document-level, character-level, and line-level operations to ensure data quality, as outlined below.

[9] https://en.wikishia.net/

3.2.1 Text-based PDFs

Text-based PDFs primarily include books and academic papers sourced from Telegram channels and Persian websites. The PDFs were converted into text using several Python libraries. To ensure quality, we tested various tools on sample documents and applied low-level heuristic filters to remove corrupted or irrelevant content.

The filtering process involves removing documents with insufficient Persian content, short text lengths, or an excessive use of symbols. This stage ensures that only relevant and high-quality documents are retained. Following this initial filtering, we apply a preprocessing pipeline to address document, character, and line-level inconsistencies, ensuring the text is properly structured. Additional technical details on these steps, including character normalization, watermark removal, and deduplication, are provided in the Appendix B.

3.2.2 Image-based PDFs (OCR)

Many papers in our dataset were converted to text using image-based OCR due to the unavailability of text-based PDFs. Given the limitations of Persian OCR, errors were introduced during text extraction. To address this, we filtered out low-quality documents, focusing on those with a high percentage of nonsensical tokens or merged words. As a result, the dataset was refined to include 321,244 documents. The documents were then processed using steps similar to those applied to web-based crawled data, with additional procedures. Additional information on the OCR-specific filtering methods is provided in the Appendix.

3.3 Social Media

Although some books and blogs may include informal Persian text or dialogues, the overall proportion of such data is minimal. The data collected from web-based sources and books generally lacks unstructured or colloquial language. Social media, however, provides a rich source of unstructured and informal linguistic data. To capture this, we gathered Persian-language data from Twitter, as well as public channels and groups from Telegram and Eitaa (an Iranian chat application). After identifying relevant channels and groups, we crawled all associated messages and processed them using the pipeline described for web-based data, with thresholds tailored to social media content. Additional processing steps we applied are outlined below.

Upon examination, we found that shorter messages were mostly replies, often lacking substantive content or containing inappropriate language. These messages were filtered out. We also identified hashtags embedded within the text and at the end of messages. Hashtags within the text were retained to preserve context, while those at the end, frequently related to political or social topics and often irrelevant to the main content, were removed. We employed regular expressions (regex) to remove channel IDs and URLs, ensuring that irrelevant content was minimized.

A notable difference in processing social media data was the deduplication strategy. We observed that many messages from different sources differed only in date or pricing—typically for goods, gold, silver, or cryptocurrencies. To address this, we removed all numeric values and dates before deduplication. After identifying and eliminating duplicate entries, we restored the original content, including numbers and dates, for the final dataset. This method ensured that informative variations were preserved while content containing no new knowledge was removed.

3.4 Final Dataset

Applying the outlined preprocessing steps, including deduplication, resulted in a significant reduction in the number of documents. As illustrated in Figure 3, the overall document count decreased by an average of **24%** after preprocessing, with a further reduction of **18.83%** following deduplication.

The largest reduction occurred in social media content, particularly from Twitter and Telegram. Many Twitter posts were short and lacked meaningful content, while Telegram messages were often redundant, brief, and became even less informative after hashtags and links were removed. The special deduplication method we applied also identified many of these messages as duplicates. Although only 1.6% of social media documents remained after processing, these retained documents were significantly longer, accounting for around 10% of the total token count from the initial data.

Image-based academic papers also experienced a considerable loss during processing. In this category, the number of documents was nearly halved, as we applied multiple criteria to remove poor-quality documents. In contrast, text-based papers saw minimal loss, with only 2% of documents eliminated during preprocessing. However, papers in this category contained more duplicates, which contributed to the reduction.

Books had the lowest proportion of document elimination during both preprocessing and deduplication. This reflects the higher quality of book content and the effectiveness of the methods used to extract data from PDF files.

For the web-crawled data, deduplication had a bigger impact than the initial preprocessing, with more documents being removed in this step. Even though we carefully tried to avoid duplicates during crawling, the nature of web crawling—often involving nested links—led to the inclusion of duplicates. Additionally, many news websites repost the same content across different agencies, which shows just how important thorough deduplication is for web-sourced data.

An interesting observation from the bar plot is that, although CulturaX FA (Nguyen et al., 2023) and Madlad-400 FA (Kudugunta et al., 2024) claim to have already undergone processing and deduplication, our language-specific preprocessing steps and content-specific deduplication further reduced their size. In Madlad-400 FA, only 7% of documents were discarded, whereas nearly 70% of CulturaX FA documents did not meet the qualifications for proper Persian data. This emphasizes the importance of language-specific processing and careful evaluation by native speakers to ensure data quality.

4 Assessing the Impact of the Matina Corpus

A large-scale Persian corpus has numerous applications in NLP, including training transformer-based models for tasks such as summarization, sentiment analysis, emotion detection, question answering, sentence embeddings, and text retrieval. Additionally, such corpora play a crucial role in pretraining large language models (LLMs) and generating instructions for LLM post-training. To assess the effectiveness of the Matina Corpus, we conducted experiments on transformer-based model training and continued pretraining of LLMs. This section provides a detailed discussion of these experiments and their outcomes.

4.1 Masked Language Model Training and Evaluation

While LLMs have excelled in various NLP tasks such as sentiment analysis and named entity recognition (NER), there remains a need for lightweight

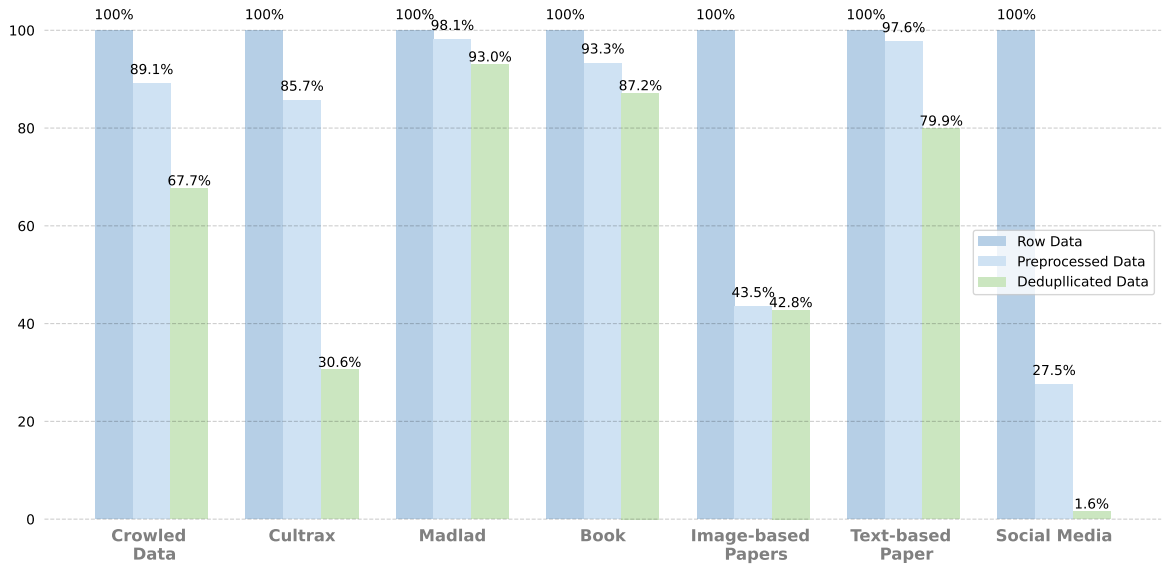

Figure 3: Data reduction during preprocessing and deduplication varies significantly across sources. Social media shows the most drastic drop, with just 1.6% of documents remaining after deduplication, while other sources retain between 56.1% and 93.3%. The three bars for each source represent the percentage of documents left after each stage. Overall, about 14% of the initial documents remain.

models that can be easily fine-tuned for specific tasks and datasets. These models are typically built on transformer-based architectures, particularly masked language models trained on large-scale datasets.

To address this need, we conducted continual pretraining of masked language models (MLMs), specifically XLM-RoBERTa Large (Conneau et al., 2020), on 54.69 billion tokens of our dataset. This extensive corpus facilitates the development of high-quality sentence embeddings, further refined by adapting the model into a Sentence-BERT architecture without Next Sentence Prediction (NSP). These enhancements yield more precise semantic representations, significantly improving Persian NLP tasks. By leveraging a well-curated dataset with rigorous preprocessing, our model effectively captures Persian linguistic nuances.

To evaluate the effectiveness of Matina corpus in training transformer models, we benchmarked out Roberta-based model against existing models using datasets such as **Arman Emo**, **Pars-ABSA**, **PQUAD**, and **PEYMA**. As shown in Table 2, our model demonstrates substantial performance gains, achieving **56.54** on **Arman Emo**, surpassing TookaBERT and AriaBERT, and **74.92** on **Pars-ABSA**, highlighting its robustness in aspect-based sentiment analysis. These results validate the impact of our dataset on enhancing Persian NLP performance, particularly within transformer-based architectures.

The success of our MLM underscores the crucial role of high-quality data in pretraining. By capturing Persian linguistic and cultural nuances, our model not only enhances task-specific performance but also advances the goal of developing inclusive and representative language technologies. This approach ensures that underrepresented languages like Persian receive the attention they deserve, fostering more equitable advancements in NLP.

4.2 Large Language Model Pretraining and Evaluation

Pretraining is essential for transferring knowledge to LLMs, shaping their linguistic and factual understanding. However, multilingual LLMs often struggle with underrepresented languages like Persian and exhibit cultural biases favoring Western perspectives (Cao et al., 2023; AlKhamissi et al., 2024) due to the dominance of English in their training data. This leads to diminished performance in other languages and cultures. Incorporating language-

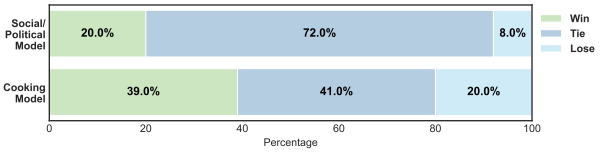

Figure 4: Win rate of pretrained models over models without pretraining.

Table 2: Results of Masked Language Models Evaluation.

Model	Arman Emo	Pars-ABSA	PQUAD	PEYMA
XLM-RoBERTa (ours)	**56.54**	**74.92**	86.82	**85.65**
TookaBERT (SadraeiJavaheri et al., 2024)	52.87	74.65	86.73	86.09
AriaBERT (Ghafouri et al., 2023)	38.23	74.59	83.14	35.78
XLM-RoBERTa (Conneau et al., 2020)	32.48	74.18	**87.6**	87.94
mBERT	6.74	68.15	85.94	65.32

Dataset	Number of tokens
Social and Politics	1.1 B
Cooking	15 M

Table 3: Number of tokens used for LLM continual pretraining. Tokens are counted by the Llama 3.1 (Dubey et al., 2024) tokenizer.

specific data during pretraining can help address this issue.

To evaluate the impact of our dataset on LLM training, we conducted the following experiment. We first tagged our dataset in an unsupervised manner using a procedure similar to InsTag (Lu et al., 2023), categorizing it into multiple domains. From these, we selected two—social and politics and cooking—and extracted a subset of data from each domain. These domain-specific subsets were then used to train models. The token count for each domain is presented in Table 3. We then constructed large instruction datasets for these domains and fine-tuned LLaMA 3.2-Instruct 8B using two different approaches: (1) continued pretraining on the domain-specific data followed by instruction tuning, and (2) direct instruction tuning without additional pretraining. To evaluate model performance, we conducted a human evaluation, where annotators ranked model outputs in a win-lose format, indicating which model provided better responses to a held-out evaluation set derived from the instruction dataset.

The evaluation results, shown in Figure 4, indicate that models benefit significantly from pretraining on even a relatively small dataset before instruction tuning. This effect is particularly noticeable in the cooking domain, where the pretrained model was preferred nearly twice as often as the model without pretraining. These findings highlight the effectiveness of the Matina Corpus in improving language models by providing high-quality, domain-specific data. Pretraining on a small, well-curated dataset not only enriches the model's knowledge but also enhances its alignment with the target language and cultural context.

5 Conclusion

In conclusion, the Matina corpus provides a crucial resource for advancing Persian NLP by addressing the limitations of existing datasets in terms of scale and diversity. With 72.9 billion tokens, it enables the training of more advanced and accurate models for tasks such as machine translation, summarization, and large-scale language modeling. We further demonstrate its effectiveness by training and evaluating transformer-based models on key NLP tasks as well as LLM pretraining, highlighting the benefits of high-quality Persian data. By making both the dataset and preprocessing tools publicly available, we aim to support further research and foster collaboration in the development of open-source tools and models for Persian.

6 Limitations

While our Persian corpus represents a significant step forward in providing high-quality data, there are several limitations to be noted:

Sub-Document Level Redundancies: Although we applied deduplication at the document level, we did not perform deduplication within documents, meaning there may be redundancies at the sentence or paragraph level. This limitation arises from the high memory and computational resources required to encode and compare sections of all documents. Unfortunately, we did not have the resources necessary to conduct this process at a finer granularity.

Sensitive Content and Language: Despite selecting Persian websites with minimal adult content and removing sensitive data from public datasets, some sensitive material and inappropriate language remain, particularly in social media data. We did not filter out offensive or explicit language, as it reflects real-world language use. However, researchers utilizing the dataset should be mindful of

this content when applying it in their work.

Residual Irrelevant Data: While we inspected samples from all data sources and employed various heuristics and filtering functions to remove irrelevant content, such as links, hashtags, advertisements, and tags, some may have evaded our processes. These elements are generally considered noise given the large scale of the dataset but may need to be addressed for more specialized use cases.

These limitations highlight potential areas for improvement, especially for projects with specific needs regarding data quality and sensitivity.

Acknowledgements

We would like to sincerely thank Mohammad Ebrahimnezhadian for his invaluable assistance in converting PDFs to usable text formats. Additionally, we would like to extend our appreciation to the National Artificial Intelligence Organization for their generous financial support, which was instrumental in the completion of this work.

References

Badr AlKhamissi, Muhammad ElNokrashy, Mai AlKhamissi, and Mona Diab. 2024. Investigating cultural alignment of large language models. *arXiv preprint arXiv:2402.13231*.

Ben Athiwaratkun, Andrew Gordon Wilson, and Anima Anandkumar. 2018. Probabilistic fasttext for multi-sense word embeddings. *arXiv preprint arXiv:1806.02901*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Andrei Z Broder. 1997. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Y Cao, L Zhou, S Lee, L Cabello, M Chen, and D Hershcovich. 2023. Assessing cross-cultural alignment between chatgpt and human societies: An empirical study. arxiv. *Preprint posted online on March*, 31.

Together Computer. 2023. Redpajama: an open dataset for training large language models.

A Conneau. 2019a. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

A Conneau. 2019b. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

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. *Preprint*, arXiv:1911.02116.

Common Crawl. 2008. Common crawl corpus. Accessed: 2024-09-28.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

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

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking nli systems with sentences that require simple lexical inferences. *Preprint*, arXiv:1805.02266.

Omid Kashefi. 2020. Mizan: A large persian-english parallel corpus. *Preprint*, arXiv:1801.02107.

Hamzeh Motahari Khansari and Mehrnoush Shamsfard. 2021. Hmblogs: A big general persian corpus. *Preprint*, arXiv:2111.02362.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2024. Madlad-400: A multilingual and document-level large audited dataset. *Advances in Neural Information Processing Systems*, 36.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, et al. 2022. The bigscience roots corpus: A 1.6 tb composite multilingual dataset. *Advances in Neural Information Processing Systems*, 35:31809–31826.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. 2020. *Mining of massive data sets*. Cambridge university press.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. # instag: Instruction tagging for analyzing supervised fine-tuning of large language models. In *The Twelfth International Conference on Learning Representations*.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *Preprint*, arXiv:1808.08745.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A Rossi, and Thien Huu Nguyen. 2023. Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. *arXiv preprint arXiv:2309.09400*.

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*.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Alec Radford. 2018. Improving language understanding by generative pre-training.

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. *Journal of machine learning research*, 21(140):1–67.

Behnam Sabeti, Hossein Abedi Firouzjaee, Ali Janalizadeh Choobbasti, SHE Mortazavi Najafabadi, and Amir Vaheb. 2018. Mirastext: An automatically generated text corpus for persian. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.

Sadra Sabouri, Elnaz Rahmati, Soroush Gooran, and Hossein Sameti. 2022. naab: A ready-to-use plug-and-play corpus for farsi. *Preprint*, arXiv:2208.13486.

MohammadAli SadraeiJavaheri, Ali Moghaddaszadeh, Milad Molazadeh, Fariba Naeiji, Farnaz Aghababaloo, Hamideh Rafiee, Zahra Amirmahani, Tohid Abedini, Fatemeh Zahra Sheikhi, and Amirmohammad Salehoof. 2024. Tookabert: A step forward for persian nlu. *arXiv preprint arXiv:2407.16382*.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*. Leibniz-Institut für Deutsche Sprache.

Targoman. 2022. Targoman dataset.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, 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*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. *Preprint*, arXiv:1804.07461.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2019. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*.

L Xue. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Appendix

A Details of Web-Based Document Processing Pipeline

A.1 Character-level Processing

Character-level processing is the initial step in our preprocessing pipeline, aimed at standardizing and cleaning the text at the most granular level. This stage involves several key operations:

1. Unicode Normalization: We convert all characters to their Persian equivalents, and remove Arabic I'rab marks. We then normalize space and tab characters to the standard keyboard space, with exceptions made for the half-space character used in specific Persian words.

2. Symbol and Number Mapping: We map symbols and numbers not belonging to the English, Arabic, or Persian character sets to their Persian equivalents using the Piraye library. This is to ensure language consistency in the dataset.

3. Repeated Characters: We identify any character repeated more than three times in sequence, typically used for emphasis, and truncate it to three occurrences to maintain readability and consistency.

4. Newline Normalization: We merge consecutive newlines, including those with spaces or tabs, to standardize line breaks across documents.

5. Non-standard Unicode Removal: By taking multiple samples from the data we found that there are chracters within the text that are not standard. We then detect and remove these non-standard Unicode characters, such as special emojis or corrupted symbols (e.g., bordered question marks) based on our predefined criteria.

A.2 Line and Paragraph-level Processing

Once character-level normalization is complete, we focus on the structural elements of the text. This stage involves:

1. HTML and JavaScript Tag Removal: We identify lines containing HTML or JavaScript tags and functions using regular expressions and replace them with newlines.

2. Custom Structures Handling: We inspected that some domains include unique tag structures that do not follow the format of standard tags (JavaScript and HTML) which are not captured by regular expressions. We identify and remove these using structures.

3. Special Character Ratio Filtering: We calculate the ratio of special characters (e.g., emojis, symbols, numbers) to total characters in each line. Lines exceeding a 0.85 ratio are removed, particularly targeting lines corrupted during text extraction, such as tables or formulas.

4. Short Line Removal: We inspected that certain sources contain incomplete or irrelevant information in the few short lines at the start of the content. We therefore remove lines shorter these specific sources.

A.3 Document-level Processing

The final stage involves document-level processing. We treat documents as a whole and remove those that meet any of the following criteria: (we refer to words as space-separated text sequences that are neither a number nor a symbol)

- Short Length Filtering: Documents shorter than 30 words are removed, as they are either corrupted or devoid of useful information.

- Non-Persian Content Removal: Documents where over 50% of characters are non-Persian are eliminated to maintain linguistic consistency and relevance.

- Repeated Words Elimination: Documents where more than 50% of the words are identical are eliminated, targeting pages that use SEO techniques or lack informative content.

- Short Lines Proportion Filtering: Documents with over 50% of lines shorter than 15 words are discarded, as they typically consist of lists or content tables.

- Out-of-Vocabulary (OOV) Words Filtering: Specifically for the CulturaX (Nguyen et al., 2023) dataset, documents containing more than 2.5% OOV words are removed to exclude irrelevant content such as code fragments or corrupted text.

Finally, we eliminate any repeated empty newlines resulted from the removal of lines or paragraphs to maintain the document's structural integrity.

A.4 Deduplication Process

To address data redundancy, we leverage the MinHash algorithm (Broder, 1997), a well-established technique for efficient similarity detection in large collections of text. The deduplication pipeline consists of the following steps:(Broder, 1997). The process involves several steps:

1. Text Normalization: We normalize Text within all documents by unifying recurring elements like days of the week and removing numbers and symbols. This normalization step is particularly crucial for content from websites that repost similar material daily. By handling these elements, we aim to reduce semantic duplicates.

2. Tokenization and Hashing: We tokenize each document into 13-grams, and hash values are computed using 128 distinct hashing functions to capture text patterns.

3. LeanMeanHash Compression: We then segment the hash values into eight sliding windows and processed using the LeanMeanHash algorithm, which compresses the hash signatures for efficient storage and comparison.

4. Graph-based Similarity Detection: Finally, we construct a graph in which each node represents a document, and edges connect nodes based on hash similarity. By identifying connected components within this graph, only one representative document per component is retained, effectively removing duplicates and near-duplicates.

This deduplication strategy ensures a significant reduction in redundant data, enhancing the corpus's quality and uniqueness, and facilitating better model generalization by preventing overfitting on repeated content.

A.5 Domain-specific Processing

Since Virgool and WikiShia domains contain highly relevant content related to Persian culture and religion, it is necessary to modify our standard preprocessing pipeline to avoid information loss. We perform the following specialized preprocessings.

For Virgool, which primarily features blog posts on diverse topics, including programming languages and mathematical content, applying the standard preprocessing thresholds resulted in the removal of valuable content. To address this, we relaxed certain filtering criteria:

- By pass the removal of numbers and symbols to preserve technical content.

- Incorporate more complex regular expressions to accurately detect and remove residual HTML tags or functions that were not filtered out by the standard pipeline.

- Adjust the ratio of Persian stopwords to lower values, and the threshold for the proportion of short lines (in relation to the total number of lines) was increased, ensuring the retention of concise but informative posts.

- Employed a privacy-preserving step to remove any personal data found in public blogs, even though the blogs are publicly accessible. This aspect of our pipeline will be discussed in detail in the subsequent section.

Another unique challenge with WikiShia was the significant presence of Arabic text, particularly due to references to the Quran and Arabic scholarly sources. To address this, we adjusted our processing thresholds: we increased the tolerance for Arabic stopwords while simultaneously lowering the threshold for Persian stopwords. This adjustment allowed us to better capture the bilingual nature of the content.

For WikiShia which includes bilingual content and presents challenges related to content duplication, we performe the following:

- Content Duplication: our recursive crawling process exposed a significant issue of content duplication. Multiple URLs often corresponded to the same page, differing only by a minor subheading. Additionally, the site includes detailed descriptions of events associated with specific dates, resulting in multiple unique URLs hosting nearly identical content tied to calendar events. To address this, we employed an exact-match deduplication strategy using MinHashLSH (Leskovec et al., 2020). Unlike our standard deduplication pipeline, we opted not to normalize or remove dates, numbers, or references to specific days of the week, as these elements are critical for preserving the chronological and cultural relevance of the content. By applying this approach, we were able to eliminate documents with a similarity threshold of 98% or higher.

- Bilingual Content Handling: Another unique challenge with WikiShia was the significant presence of Arabic text, particularly due to references to the Quran and Arabic scholarly

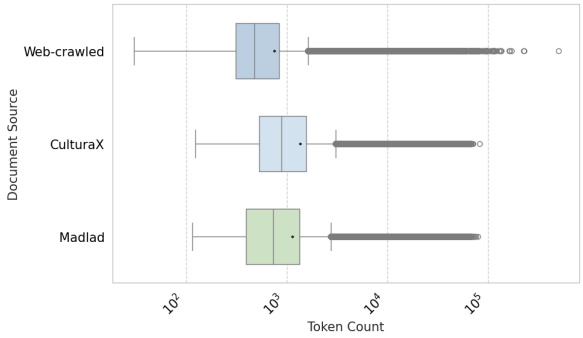

Figure 5: Document Length Distribution For Web-based Crawled Data

sources. To address this, we adjusted our processing thresholds. The tolerance for Arabic stopwords was increased, while the threshold for Persian stopwords was lowered, effectively capturing the bilingual nature of the content.

The boxplot in Figure 5 illustrates the token count distribution across three document sources we had for web-based data. The results are in tokens by the Llama3.1 (Dubey et al., 2024) and after the application of our comprehensive preprocessing pipeline and deduplication. Data crawled by the team, named Web-crawled, show the widest range, with a median around 1000 tokens and some documents extending beyond 10^5 tokens. Madlad exhibits a slightly narrower distribution but still maintains substantial variation. CulturaX demonstrates the most compact distribution, with a lower median and maximum token count. These distributions highlight the success of our preprocessing in maintaining diversity while standardizing document lengths. The presence of outliers, particularly in the Web-crawled and Madlad sources, indicates that our pipeline preserves some longer, potentially information-rich documents. This final data composition ensures a balance between consistency and variety, crucial for robust model training and generalization.

B Details of Book and Paper Processing Pipeline

For data extraction and OCR conversion, we utilized a range of Python libraries, including Selenium[10], BeautifulSoup[11], and Pytesseract[12]. Text-based PDFs were converted using lightweight tools such as pdf2image[13], while image-based PDFs required more advanced processing with Pytesseract and Fitz[14]. To improve accuracy, we employed an iterative approach, applying multiple tools to the same documents and manually inspecting those with errors before refining the extraction process.

B.1 Text-based PDFs: Detailed Processing

After removing corrupted or non-Persian documents, we apply a 3-stage processing pipeline involving document-level, character-level and line-level processing. Unlike documents from web, we first apply the document-level processing to avoid redundant processing.

B.1.1 Document-level Processing

In the first stage, we applied document-level processing, where a document was viewed holistically. If it met any of the following criteria, it was eliminated:

- Documents with fewer than 150 space-separated words.
- Documents containing less than 50% Persian characters.
- Documents with an average word length of fewer than 3 characters or greater than 10 characters.
- Documents with a numeric or symbolic character ratio exceeding 0.8.
- Documents where over 80% of the lines were considered short, defined as containing fewer than four space-separated words.
- Documents where fewer than 10% of the words were Persian or Arabic stopwords.

B.1.2 Character-level Processing

Given that many of the books contained long Arabic text, which needed to be preserved, we only normalized non-Arabic, non-English, and non-Persian characters and symbols to their Persian format. We did not remove I'rab (diacritics). Standard procedures, such as replacing consecutive repeated characters, normalizing newlines, and removing non-standard Unicode characters, were applied as in previous section, though with additional Unicode characters added to the filtering set. Furthermore,

[10] https://selenium-python.readthedocs.io/
[11] https://beautiful-soup-4.readthedocs.io/en/latest/
[12] https://github.com/madmaze/pytesseract
[13] https://github.com/Belval/pdf2image
[14] https://github.com/pymupdf/PyMuPDF

nonsensical patterns detected in the text, which added no value and increased noise, were removed. These patterns included:

- Website links to the source of the document.
- Repeated occurrences of the book's title at the top or bottom of pages.
- Page numbers in various forms, such as صفحه ۱, صفحه۱, ص۱, صفحه۱ از ۰۰۲, ۱ از کتاب ... , etc.
- Tags related to cover pages.
- Errors or tags related to multimedia, such as `'Your browser does not support the audio tag.'`
- Images or tables converted to 'UNKNOWN' strings.
- Personal information, such as phone numbers, email addresses, account numbers, and credit card numbers (e.g., Shaba numbers), which were found at the end of some books and at the beginning of papers.

B.1.3 Line-level Processing

Following character-level processing, we performed line-level processing to remove lines that contained formulas or tables that were corrupted during the conversion from PDF to text. As part of this stage, the following types of lines or paragraphs were removed:

- Lines with a numeric character ratio exceeding 0.8.
- Lines with a symbolic character ratio exceeding 0.8.
- Lines that were repeated multiple times within the document, which often included hidden watermarks or the repeated mention of the book's title.

B.1.4 Deduplication

To avoid redundancy, a deduplication process was applied using MinHash and Locality-Sensitive Hashing (LSH). We deduplicated documents within each source, ensuring that only unique documents were retained.

B.2 Image-based PDFs (OCR): Detailed Processing

For OCR-processed documents, the primary issue was the introduction of errors during text extraction. To mitigate this, we employed the following steps:

1. Removed content preceding the keywords section, which was often corrupted, using regex patterns to detect specific document structures.
2. Removed documents with more than 5% out-of-vocabulary tokens.
3. REmoved papers containing more than 10 words exceeding 15 characters, indicative of merged words.

Although some OCR-generated text still contains minor issues, such as occasional word merging, these are manageable with model tokenizers and do not significantly affect overall context and understanding.

The boxplot in Figure 6 shows the token count distribution across different document sources. Books have a notably higher median token count and broader range compared to papers. Both image-based and text-based papers display lower token counts with numerous outliers, indicating diverse token lengths. Text-based papers have a lower median as they contain paper summaries as well as internal papers. Image-based papers also contain high-quality and longer scientific documents.

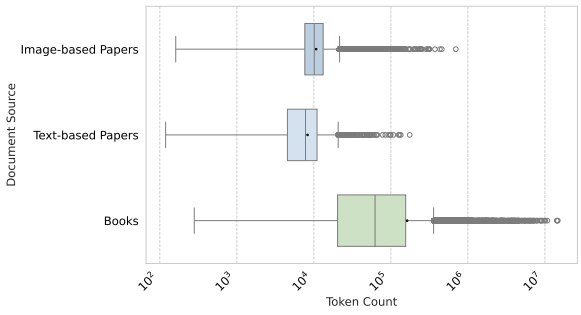

Figure 6: Document Length Distribution For Crawled Books and Papers