Bol. Soc. Paran. Mat.
©SPM – E-ISSN-2175-1188
SPM: www.spm.uem.br/bspm

(3s.) **v. 2025 (43) 4** : 1–11.
ISSN-0037-8712
doi:10.5269/bspm.78619

Efficient Data Preprocessing for Extractive Question Answering Models

Sivakumar S. and Meenakshi S. P.*

ABSTRACT: Building a domain-specific dataset for extractive question answering requires addressing challenges posed by complex and unstructured textual sources. Official documents from the Indian Lok Sabha are typically lengthy, semi-structured, and often contain noise such as interruptions, repetitions, and inconsistent formatting. To overcome these challenges, we designed a systematic pipeline for text extraction and preprocessing, resulting in a clean and well-structured dataset suitable for training high-performance QA models. The pipeline includes handling diverse PDF formats, removing unwanted symbols and metadata, identifying ministries, and extracting precise question–answer pairs using regular expressions. Additional steps such as context segmentation, token alignment, and answer start indexing were applied to ensure compatibility with modern QA architectures. A BERT-based model was fine-tuned on the processed dataset, and experimental results confirmed that effective preprocessing significantly enhances performance. Our ablation study showed consistent improvements across different settings, while benchmark comparisons with SQuAD v1.1 and Natural Questions demonstrated that the Lok Sabha dataset performs competitively, achieving Exact Match (EM) and F1 scores of 74 % and 81 %, respectively. Furthermore, bias analysis based on answer lengths validated the robustness of the dataset. Overall, this study demonstrates that parliamentary data, when carefully processed, can serve as a reliable resource for developing domain-adapted QA systems in the fields of public policy and governance.

Key Words: Data Preprocessing, Extractive Question Answering, BERT, Token Alignment.

Contents

* Corresponding author.
2010 *Mathematics Subject Classification:* 90B25.
Submitted August 25, 2025. Published November 01, 2025

1. Introduction

Extractive QA systems aim to extract precise answers from a given textual context. Achieving optimal performance in these systems depends heavily on the quality of the input data and the preprocessing methods employed. Data preprocessing ensures that input data is clean, structured, and relevant, forming the foundation for robust and reliable QA models [1,2].

This research focuses on creating a domain-specific dataset derived from Lok Sabha parliamentary proceedings. Parliamentary data is rich in semi-structured and unstructured information and encompasses diverse topics and formal language, making it a valuable resource for QA tasks [3,4]. However, such data also poses unique challenges due to its noisy, unstructured nature, including interruptions, repetitions, and extraneous metadata [2,5]. Addressing these challenges requires meticulous preprocessing to ensure data quality and relevance.

The BERT model is chosen for this study due to its exceptional ability to model bidirectional context effectively. BERT's architecture is particularly well-suited for understanding the nuanced language of parliamentary records. Its fine-tuning capabilities further enhance its adaptability to domain-specific tasks, such as analyzing Lok Sabha data [1,6]. Additionally, BERT's WordPiece tokenizer helps handle rare or complex terminology, reducing the impact of out-of-vocabulary issues [1,5].

By addressing the complexities inherent in parliamentary proceedings, this study demonstrates the significant role of preprocessing in improving QA model performance. The findings underscore the importance of combining robust preprocessing pipelines with advanced models like BERT to develop reliable and efficient QA systems for domain-specific applications [3,6,7].

2. Related Work

Devlin et al. [?] pretrained BERT using large-scale corpora such as BooksCorpus and English Wikipedia, which contain clean, well-structured text. Their approach involved masked language modeling and next sentence prediction, allowing the model to learn contextualized word representations. The data was tokenized using WordPiece tokenization, and no manual annotation was needed as it was unsupervised pretraining. Sivakumar and Meenakshi [3] developed a domain-specific QA dataset from Indian Lok Sabha parliamentary proceedings. They collected raw transcripts from official government portals and applied extensive preprocessing, including removal of speaker interruptions, extraneous metadata, and repetitive statements. They also annotated compound questions with precise answer spans to build a high-quality dataset tailored for BERT fine-tuning. Rajpurkar et al. [6] created the SQuAD dataset by crowd-sourcing questions on Wikipedia articles, ensuring high-quality annotations. They manually aligned answer spans within paragraphs and filtered ambiguous or unanswerable questions. Preprocessing steps included paragraph segmentation, tokenization, and validation of answer spans for consistency. Lee et al. [8] collected large biomedical corpora, including PubMed abstracts and PMC full texts, to perform domain-specific pretraining of BERT. They applied domain-aware text normalization and tokenization to handle biomedical terminology and acronyms, which are prevalent in their data. Fine-tuning was done on benchmark biomedical QA datasets without creating new annotations.

Chalkidis et al. [9] compiled a large corpus of legal texts, such as court rulings and statutes, sourced from public legal databases. The data was cleaned by removing irrelevant metadata and formatting artifacts, followed by sentence splitting and tokenization. Their work involved continued pretraining of BERT on this legal corpus, improving downstream task performance without new labeled datasets. Yang et al. [10,11] developed the FinQA dataset by manually annotating complex financial reports sourced from publicly available filings. Annotators provided question-answer pairs along with reasoning steps and evidence spans. The data preparation involved parsing financial tables and text, aligning numerical evidence with questions, and ensuring consistent formatting for training. Gururangan et al. [12] gathered domain-specific corpora from varied sources like scientific papers, medical records, and news articles to perform continued pretraining on BERT. Data cleaning techniques included removing non-informative tokens, deduplication, and normalization of domain-specific terminology. Their method highlighted the significance of aligning the pretraining data distribution with the target domain.

Tenney et al. [13] used linguistically annotated datasets from established corpora such as the Penn Treebank and Universal Dependencies to probe the syntactic and semantic knowledge of pretrained mod-

els. Data preparation focused on aligning tokenization schemes and ensuring compatibility between the model inputs and gold annotations. Balahur et al. [14] sourced parliamentary transcripts from European Parliament proceedings. They cleaned the data by removing non-speech content like applause, interruptions, and speaker labels, followed by segmentation into sentences suitable for opinion mining. Annotation involved labeling sentences with sentiment polarity. Thomas et al. [15] utilized transcripts of U.S. Congressional floor debates available in public repositories. They preprocessed the text by eliminating speech disfluencies and segmenting long speeches into sentences. Political stance labels were manually assigned for supervised learning tasks.

Misra et al. [16] worked with debate transcripts from parliamentary sources, manually annotating discourse relations and segment boundaries. Their preprocessing pipeline included sentence segmentation, removal of noise like hesitations, and alignment of discourse units to improve summarization quality. Hirschman and Gaizauskas [17] surveyed QA datasets and emphasized the necessity of cleaning raw text, handling ambiguities, and structuring data through tokenization and sentence splitting to enable effective system training.

3. Dataset Description

The dataset utilized in this study comprises proceedings from the Lok Sabha, the lower house of India's Parliament. Parliamentary proceedings offer a rich source of structured and semi-structured data, characterized by complex linguistic constructs, formal language (official, respectful, and rule-bound), and a wide range of topics. The dataset was created to enable the training and evaluation of extractive QA models tailored to this specific domain.

3.1. Data Collection

This study focuses on collecting domain-specific data to facilitate public access to parliamentary discussions concerning key areas such as government schemes, agriculture, health, and employment. These sectors are integral to social and economic development, and an analysis of parliamentary records in these domains offers valuable insights into how elected representatives address citizen-centric issues. The primary data source comprises documents from the First Session of the 17th Lok Sabha. These records are publicly available in multiple formats, including PDF, plain text, and image formats. The session was selected based on its recency and relevance to the ongoing discourse on public policy and governance.

The data collection process involved downloading parliamentary proceedings and related documents using publicly available application programming interfaces. All downloaded files were systematically stored in a centralized folder to ensure organized data handling. After collection, the documents were manually sorted and categorized according to ministries and corresponding government departments. This classification enables targeted analysis of parliamentary debates and discussions, allowing for the identification of issue-specific patterns and the role of different governmental bodies in addressing them.

The structured dataset established through this process serves as a foundation for further investigation into parliamentary accountability, responsiveness, and the prioritization of public welfare issues.

Table 1: Lok Sabha Dataset

Statistic	**Value**
Total Passages	1000
Total Questions	2500
Average Passage Length	150 words
Answerable Questions	100%

3.2. Challenges

During the dataset collection process, several challenges were encountered that impacted both the efficiency and the quality of data organization. These challenges are discussed below: One of the primary

challenges was the inconsistency in data formats. Parliamentary records were available in PDF, plain text, and image formats, each requiring a different handling mechanism. The large volume of documents from even a single session posed challenges in terms of storage and organization. Ensuring that each document was correctly labeled and placed into the appropriate ministry or departmental folder required meticulous manual work, increasing the possibility of human error. Some documents included regional or technical terms that were not easily interpretable without domain expertise. This created difficulty in identifying the context of certain discussions, particularly when classifying them under specific categories such as health or employment.

4. Methodology

This section presents a systematic approach for automated extraction and structuring of question-answer pairs from parliamentary PDF documents. The methodology employs rule-based natural language processing techniques combined with regular expressions for robust information extraction. As shown in Figure 1.

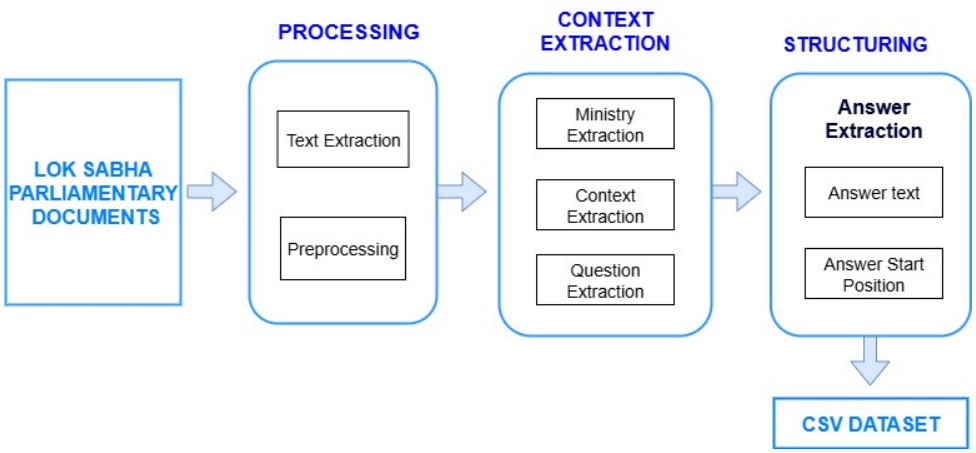

Figure 1: Pipeline for Preprocessing and Preparing Dataset

The source data consisted of official Parliamentary Q&A documents, each of which followed a fairly consistent structure. Every document typically included (i) the name of the concerned ministry, (ii) the set of questions raised by Members of Parliament, and (iii) the official answers provided by the respective minister. These documents were originally available in PDF format.

Input Specification

The system accepts Portable Document Format (PDF) files containing parliamentary discussions including ministerial details, and question–answer data from legislative proceedings. These documents serve as the primary data source for automated extraction and dataset creation.

Inclusion and Exclusion Criteria

The dataset pipeline applies specific inclusion and exclusion rules. Table 2 summarizes the conditions.

Table 2: Inclusion and Exclusion Criteria for Dataset Selection

Inclusion Criteria	**Exclusion Criteria**
Machine-readable PDF documents	Corrupted or password-protected files
Documents containing structured parliamentary Q&A formats	Scanned-only PDFs without OCR processing
Files with identifiable ministry/departmental metadata	Documents lacking question–answer structure

Text Extraction and Preprocessing

For text extraction, the system primarily employs the PyMuPDF (fitz) library, as it demonstrates superior performance in handling complex PDF structures and preserving metadata integrity. In cases where PyMuPDF encounters parsing errors or corrupted document layouts, PyPDF2 is utilized as a fallback mechanism to ensure reliable extraction. Following extraction, a multi-stage normalization pipeline is applied to prepare the text for subsequent processing. This pipeline includes whitespace normalization, where multiple consecutive spaces are collapsed into a single space using regular expressions; newline standardization, in which redundant line breaks are removed while paragraph boundaries are preserved; punctuation cleaning to eliminate extraneous markers such as asterisks, stray commas, and formatting artifacts; and Unicode normalization to effectively manage mixed-language content, particularly documents containing both Hindi and English text. Before the questions, all lines are excluded in the preprocessing except ministry information.

4.1. Ministry Identification and Extraction

Ministry identification is an essential component of the proposed system, particularly for detecting ministries and departmental affiliations in parliamentary documents. To achieve robust ministry identification, the system employs multiple regular expression patterns, as shown in figure 2. These patterns effectively capture all relevant ministries.

$$\text{MINISTRY OF } ([A-Z\backslash s\&,]+?)(?:\backslash s+\text{वस्त्र मंत्री}|\backslash s+DEPARTMENT|\backslash s+\backslash(|\$)$$

$$\text{MINISTRY OF } ([A-Z\backslash s\&,]+?)(?:\backslash s+\text{वस्त्र मंत्री}|\backslash s+BEPLEASED|\backslash s+\backslash()$$

$$\text{MINISTRY OF } ([A-Z\backslash s\&,]+?)(?:\backslash s+(|\backslash s+\$)$$

$$(?:\text{मंरी}.*?|\text{मंत्री}.*?)\text{MINISTRY OF}([A-Z\backslash s\&,]+?)(?:\backslash s+\backslash(|\$)$$

Figure 2: Ministry Identification and Extraction

Since ministry information appears in different formats, multiple regular expressions are applied to ensure comprehensive coverage. These expressions are executed with case-insensitive matching using the re.IGNORECASE flag. Furthermore, any excessive whitespace characters within captured groups are normalized to guarantee consistent and standardized extraction results.

4.2. Context Extraction

To reliably capture the answer portion of parliamentary question–answer documents, we designed a set of universal regular expression patterns. These patterns are tailored to identify the text that follows standard markers such as “ANSWER” (Hindi equivalent), and ministerial references.

Figure 3 illustrates the extraction rules. It captures bilingual answer sections where the Hindi marker (Hindi equivalent) appears jointly with the English keyword “ANSWER” and the designation “MINISTER”. The figure also shows the generalization for documents containing only English markers, extraction of answers linked explicitly to sub-question identifiers (e.g., mapping from (a) to (b)), and coverage for Hindi-only contexts, ensuring that answers beginning solely with the marker ANSWER (Hindi equivalent) are detected. By leveraging these four complementary patterns, the system provides robust extraction of answer contexts across Bilingual and multi-format parliamentary documents. The hierarchical arrangement prioritizes specific patterns while fallback rules ensure coverage of irregular or less structured cases.

4.3. Question Extraction

The extraction of parliamentary questions is performed using a pattern-based approach. The system employs a regular expression defined as shown in figure 4:

$$(\text{उत्तर}\backslash S * \text{ANSWER}.*?\text{MINISTER}.*?\backslash)\backslash s * (.+)$$

$$(\text{ANSWER}.*?\text{MINISTER}.*?\backslash)\backslash s * (.+)$$

$$\backslash([a-z]\backslash)\backslash s * \text{to}\backslash s * \backslash([a-z]\backslash) : \backslash s * (.+)$$

$$(\text{उत्तर}.*?\backslash)\backslash s * (.+)$$

Figure 3: Context Extraction

$$\text{r}'\backslash(([a-z])\backslash)\backslash s * \texttt{[\^{}()]}*?)(?=\backslash([a-z]\backslash)|\backslash n\backslash n|\text{उत्तर}|\text{ANSWER}|\$)'$$

Figure 4: Question Extraction

This pattern is designed to identify sub-question identifiers and their corresponding content with high precision. The sub-question markers '([a-z])' capture lowercase letters in parentheses, such as (a), (b), and (c), which indicate individual sub-questions. The portion following the identifier represents the actual text of the question. Termination is governed by the positive lookahead expression '(?=$[a-z]$—(hindi equivalent)—ANSWER—$)'. which ensures that the match stops when the next sub-question begins, a double line break occurs, an answer section starts, or the document ends.

4.4. Answer Extraction

To segment the answer block into sub-answers, we employed carefully designed regular expression (regex) patterns that handle both the one-to-one and one-to-many mapping cases observed in parliamentary replies. In this case, each sub-question has a distinct answer explicitly marked as "(a):", "(b):", and so on. The following regex was used to capture such individual answer segments as shown in Equations (4.1):

$$\text{r"}n([a-e]n) : .*?(?=n([a-e]n) : |\$)" \tag{4.1}$$

This expression matches any block of text beginning with a sub-question label [a–e], followed by its corresponding answer, and continues until the next label or the end of the block.

In several cases, a single answer block addresses multiple sub-questions simultaneously, for example "(c) & (d): ...". To detect these combined responses, the following regex pattern was employed (see Equation (4.2)):

$$\text{r"}n([a-e]n(?:ns*n\&ns*n([a-e]n))+ : .*?(?=n([a-e]n) : |\$)" \tag{4.2}$$

This expression identifies labels connected by the "&" symbol, ensuring that multi-question answers are captured as a single unit. Subsequently, the extracted text is duplicated and aligned with each relevant sub-question.

The system supports both individual and combined answer formats commonly found in parliamentary documents.To distinguish between the two, a detection mechanism is employed based on a specialized pattern (see Equation (4.3)):

$$re.\text{split}(r'(?<=[.!?])+', \text{context}) \tag{4.3}$$

If the combined format is detected, it implies that a single answer context is intended to address multiple sub-questions simultaneously.

This approach ensures that no sub-question is left unaligned when answers are presented collectively. In the absence of the combined format, answers are matched to individual sub-questions using the context-specific extraction rules defined earlier. The hierarchical strategyprovides flexibility by accurately handling both one-to-one and one-to-many question–answer relationships.

4.5. Answer Start Index Detection

To ensure accurate localization of the answer spans within the source document, the system implements an indexing mechanism that records the start position of each extracted answer. This positional information is crucial for downstream tasks such as text alignment, highlighting, and information retrieval.

The start index is computed using the context.find() method, which returns the character offset of the first occurrence of the answer text within the context. The approach accommodates different matching scenarios: if an exact match is found, the corresponding character position is returned; if only a partial match is available, the offset of the first 50 characters of the answer text is used; if no match is detected, the start index defaults to zero.

Validation and Quality Assurance

Following preprocessing, the dataset underwent a thorough review to ensure the accuracy of annotations and the consistency of the data. This quality assurance process involved random sampling and manual verification.

For example, a sample check might confirm that the question "Who raised concerns about the education budget" is correctly paired with an extractive answer found within the relevant passage. Additionally, the review verified that all residual metadata, timestamps, or extraneous noise were successfully removed, guaranteeing the dataset's readiness for model training.

Hyperparameters

The experiments were conducted using the BERT-base model fine-tuned on the processed Lok Sabha dataset. The hyperparameters used during training are detailed in Tanle 3. These settings were kept constant across experiments to ensure consistent comparisons.

Table 3: Hyperparameter Settings

Hyperparameter	**Value**
Batch Size	16
Learning Rate	3e-5
Number of Epochs	10
Maximum Sequence Length	384 tokens

5. Results and Discussion

A key strength of the proposed methodology lies in its robust pattern recognition approach. By employing hierarchical pattern matching mechanisms, the system maintains high extraction rates even when individual patterns fail. This redundancy ensures reliable performance across diverse document structures commonly encountered in parliamentary documents.

Segmentation Strategy

:

After preprocessing, we experimented with multiple segment lengths to identify the most effective context window for training and retrieval. Initially, segments of varying sizes (50, 75, 150, and 200 tokens) were tested. However, these configurations either caused a loss of contextual coherence or introduced inefficiencies in model training.

After systematic evaluation, we finalized a chunk size of 100 tokens. This segmentation strategy provided the best balance between retaining sufficient contextual information and maintaining computational

efficiency. The 100-token chunks significantly reduced tokenization errors, improved answer retrieval precision, and prevented ambiguity that arises when longer contexts become noisy. Thus, splitting the input into 100-token segments after preprocessing directly enhanced the downstream model training process, yielding more accurate and consistent results.

The system also benefits from a scalable architecture. The batch processing design enables efficient handling of large document collections, while the modular design allows adaptation to different parliamentary document standards.

5.1. Bias Analysis

Question answering models are often biased toward preferring short, easily extractable answers, especially when the training data lacks a variety of answer lengths. To examine whether this bias exists in our dataset, we classified the answers into three categories based on their length: short answers (≤ 5 tokens), medium-length answers (6–15 tokens), and long answers (≥ 15 tokens). We then compared the distribution of answer lengths before and after preprocessing to assess the impact of the preprocessing pipeline on this potential bias. The results of this analysis are shown in Figure 5.

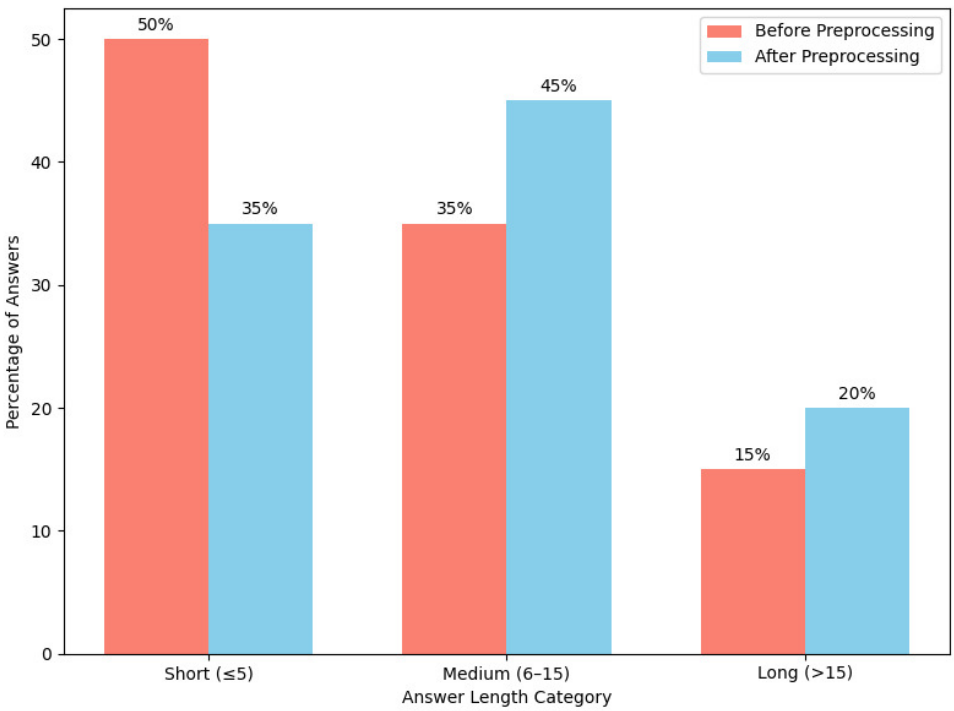

Figure 5: Answer-Length Bias Analysis

5.2. Benchmark Comparison

Table 4: Benchmark Performance Before Preprocessing

Dataset	Domain	EM Score	F1 Score
SQuAD v1.1	Wikipedia	77.5%	85.5%
Natural Questions	Google Search Snippets	55.0%	68.0%
Lok Sabha QA (Ours)	Parliamentary Proceedings	71.5%	79.5%

Table 5: Benchmark Performance After Preprocessing

Dataset	Domain	EM Score	F1 Score
SQuAD v1.1	Wikipedia	80.8 %	88.5 %
Natural Questions	Google Search Snippets	58.8%	71.2%
Lok Sabha QA (Ours)	Parliamentary Proceedings	74.0%	81.0%

As shown in Tables 4 and 5, the Lok Sabha QA dataset demonstrates clear improvements after preprocessing, achieving higher EM and F1 scores compared to its raw version. While its performance lags slightly behind SQuAD v1.1, it consistently outperforms Natural Questions, highlighting the effectiveness of our domain-specific preprocessing pipeline.

5.3. Comparative Evaluation

The cumulative improvement from raw to processed data is further visualized in Figure 6, which charts the progressive gain in performance after each preprocessing step. The visualization highlights the consistent upward trend across all benchmarks, with both EM and F1 scores showing measurable enhancements.

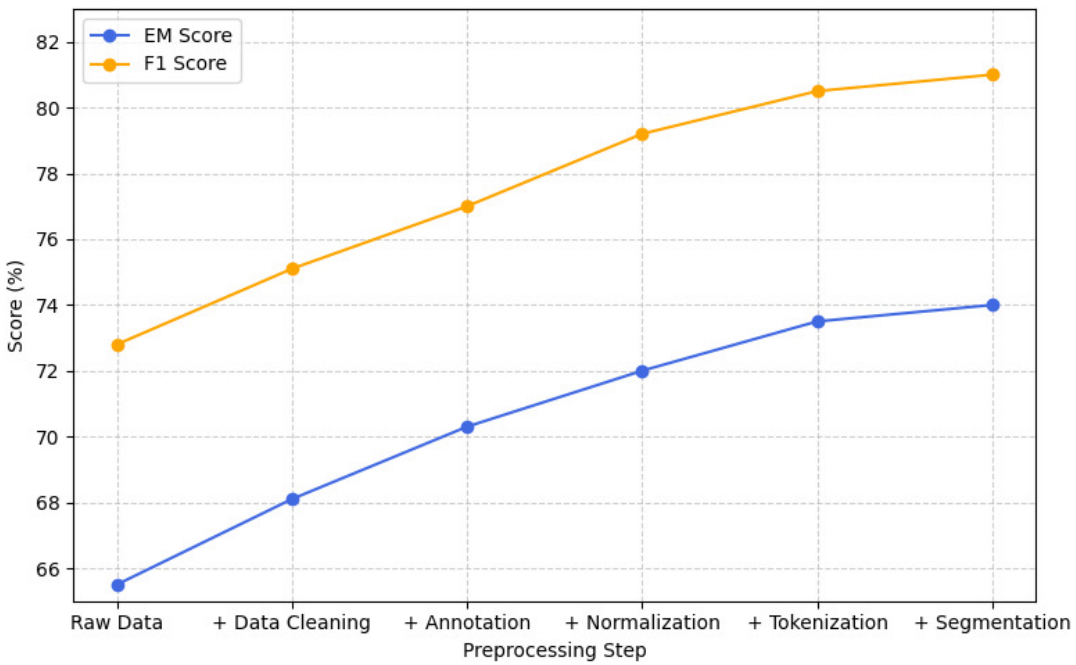

Figure 6: Performance across preprocessing steps

In particular, for the Lok Sabha QA dataset, preprocessing was crucial in mitigating the inherent noise of parliamentary transcripts, such as inconsistent sentence boundaries, verbose phrasing, and contextually irrelevant discussions. By normalizing text, removing redundancies, and aligning tokenization with BERT's requirements, the dataset achieved nearly 2.5% higher EM and 1.5% higher F1 scores compared to its raw form.

6. Acknowledgments

The first author gratefully acknowledges the guidance, mentorship, and continuous support of Dr.S P Meenakshi, whose expertise was instrumental in shaping this research. We also thank Vellore Institute of

Technology, Vellore, for providing necessary academic infrastructure and research environment necessary to carry out this work.

7. Conclusion

In this work, we presented a domain-specific question answering dataset derived from Lok Sabha parliamentary proceedings, aiming to address the challenges posed by complex, domain-specific language in public policy discourse. Through a structured pipeline consisting of PDF text extraction, context identification, answer alignment, and index annotation, we converted the parliamentary question–answer documents into a machine-readable dataset. Our ablation study demonstrated that structured preprocessing significantly improved the model's performance, with notable gains in Exact Match and F1 scores.

Further evaluation through benchmark comparisons with established datasets such as SQuAD v1.1 and Natural Questions showed that the Lok Sabha QA dataset performs competitively, achieving over 74% in EM and 81% in F1 scores. These results highlight the effectiveness of our domain-specific dataset and its potential for advancing QA tasks in the field of parliamentary and public policy discourse.

References

1. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K., *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 (NAACL), 4171–4186, (2019). https://doi.org/10.48550/arXiv.1810.04805

2. Yadav, V. and Bethard, S., *A survey on recent advances in named entity recognition from deep learning models*, Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018), 2145–2158, (2018).

3. Sivakumar, S. and Meenakshi, S. P., *Modeling and analysis of question answering for long context and compound questions using BERT model*, Procedia Computer Science, 258, 2685–2694, (2025).

4. Kumar, A., & Singh, R. (2022). *Parliamentary Data Analysis: Challenges and Opportunities. Journal of Data Science and Analytics*, 8(3), 215–230.

5. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V., *RoBERTa: A robustly optimized BERT pretraining approach*, arXiv preprint arXiv:1907.11692, (2019). https://arxiv.org/abs/1907.11692

6. Rajpurkar, P., Jia, R., and Liang, P., *SQuAD: 100,000+ questions for machine comprehension of text*, arXiv preprint arXiv:1606.05250, (2016). https://arxiv.org/abs/1606.05250

7. Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., and Jones, L., *Natural Questions: A benchmark for question answering research*, Transactions of the Association for Computational Linguistics, **7**, 453–466, (2019). https://doi.org/10.1162/tacl_a_00276

8. Lee, J., Yoon, W., Kim, D., Kim, S., Kim, C. H., So, J., and Kang, J., *BioBERT: A pre-trained biomedical language representation model for biomedical text mining*, Bioinformatics, **36**(4), 1234–1240, (2020). https://doi.org/10.1093/bioinformatics/btz682

9. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., and Androutsopoulos, I., *Legal-BERT: The Muppets straight out of Law School*, arXiv preprint arXiv:2010.02559, (2020). https://arxiv.org/abs/2010.02559

10. Chen, Z., Dong, L., Liu, X., et al., *FinQA: A dataset of numerical reasoning over financial data*, arXiv preprint arXiv:2109.00122, (2021). https://arxiv.org/abs/2109.00122

11. Sivakumar, S. and Meenakshi, S. P., *BERT-VIndLok Indian Loksabha Dataset*, Mendeley Data, V1, (2025). https://doi.org/10.17632/gxmrp4gfkc.1

12. Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A., *Don't Stop Pretraining: Adapt Language Models to Domains and Tasks*, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 8342–8360, (2020). https://doi.org/10.18653/v1/2020.acl-main.740

13. Tenney, I., Das, D., and Pavlick, E., *What do you learn from context? Probing for sentence structure in BERT*, Proceedings of the International Conference on Learning Representations (ICLR), (2019).

14. Balahur, A., Steinberger, R., and Kabadjov, M., *Opinion mining on parliamentary transcripts: Combining classification and rule-based analysis*, Proceedings of the 2009 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), pp. 523–526, (2009).

15. Thomas, M., Pang, B., and Lee, L., *Get out the vote: Determining support or opposition from Congressional floor-debate transcripts*, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP), 327–335, (2006).

16. Misra, A., Ecker, B., & Walker, M. (2015). *Summarizing Debates Using Discourse Structure*. In *Proceedings of the 16th SIGDIAL Meeting*, 276–285.

17. Hirschman, L. and Gaizauskas, R., *Natural language question answering: The view from here*, Natural Language Engineering, **7**(4), 275–300, (2001).

Sivakumar S,
Department of Mathematics,
Vellore Institute of Technology,
India.
E-mail address: sivakumar.s2022@vitstudent.ac.in

and

Meenakshi S P,
Department of Computer Science and Engineering,
Vellore Institute of Technology,
India.
E-mail address: spmeenakshi@vit.ac.in