Comparative Analysis for Requirement Classification Using Transformer based Pre-trained Models for Digital Governance RFPs

Manisha Tiwari*, Shagun Srivastava and Padmaja Joshi
Department of Computer Engineering, MPSTME, NMIMS University, Vile Parle, Mumbai, 400056, Maharashtra, India
C-DAC, Gulmohar Cross Road no 9, Juhu, Mumbai, 400049, Maharashtra, India
E-mail: manisha.tiwari@nmims.edu

Keywords: digital governance, transfer learning, requirement classification, request for proposal document, functional requirements, non-functional requirements

Received: June 2, 2025

Digital Governance deals with vast data. One of the domains in digital governance is tendering, which uses multiple documents, including a Request for Proposal (RFP). These documents contain extensive information, particularly detailing the project's expected requirements. For these RFP documents, the identification and classification of requirements are essential. One of the existing datasets under the software engineering domain is the PROMISE dataset for software requirements; this work takes inspiration from the PROMISE dataset and curates a dataset for digital governance software development-related RFP documents. The curated domain-specific dataset for text classification uses a pre-trained language model to classify functional and non-functional requirements. Experiments were performed to compare the Transformer's model performance with the baseline dataset, the curated DigiGov RFP dataset, and the concatenated PROMISE + DigiGov RFP datasets.

The model's statistical performance across the datasets is assessed using an ANOVA test. The work focuses on automating RFP document statement classification using transformer-based pre-trained models through transfer learning, increasing productivity and accuracy in the field of digital governance. The research shows that using state-of-the-art techniques for RFP documents can effectively enhance the quality of the bidding process. This technique can bring automation to requirement analysis in the bidding process, strengthening the digital governance process.

Povzetek: Študija pokaže, da lahko Transformerski modeli z uporabo prenosnega učenja učinkovito avtomatizirajo razvrščanje zahtev v RFP dokumentih ter izboljšajo natančnost in učinkovitost digitalnega upravljanja.

1 Introduction

Digital governance uses state-of-the-art technologies to improve the delivery of government services to users [1]. The main objective of digital governance is to provide government services in an easy-to-use manner with 24/7 availability. In addition, the services offered should be transparent and efficient. To attain this goal, through e-Tendering and procurement, government departments try to identify the agencies that will implement the desired services effectively and on time. e-Tendering and procurement is an online platform for conducting government tendering and procurement. Many documents like Tender notices, Requests for Proposals (RFP), Requests for Quotations (RFQ), Vendor Profiles, Financial Bids/Proposals, etc., are published in this domain. Government departments and agencies use government electronic tendering services, inviting tenders for required goods, services, and products. Bidders carefully review the tender documents to understand the requirements and then submit their bids online. Figure 1 illustrates the steps of the online tendering process.

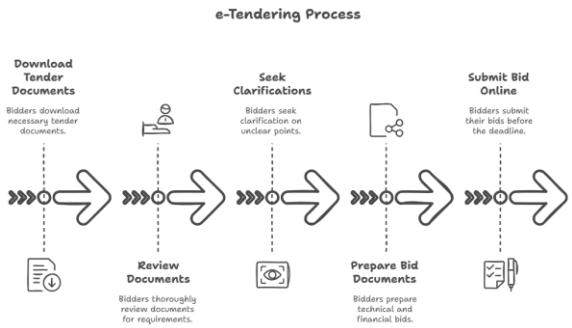

Figure 1: e-Tendering process

An organisation initiates the tendering process for a project for goods or services using an RFP. Many government and non-government organisations use these documents to

solicit proposals from vendors or suppliers for services/solutions. These documents usually contain information like the scope of the project, technical and functional requirements of the projects, expected timeline, guidelines, and criteria for who can participate in the bidding process. These RFP documents aim to clarify all the project requirements so that bidders can understand and prepare the detailed proposal accordingly. Bidders play a crucial role in the bidding process. The process used by bidders includes understanding the requirements mentioned in the RFP document by reviewing it thoroughly and, after getting sufficient clarification, preparing a technical and financial proposal that mentions how they plan to meet the requirements and at what cost, adhering to the guidelines they submit the bids. Bidders usually manually analyze RFP documents, where human expertise is required to identify and understand the requirements and decide whether they are feasible for the organization.

With the advent of machine learning and AI, researchers and practitioners explore the technology to simplify the required understanding of RFPs. Specific works of literature have attempted to study the role of automating the information classification/extraction of data from the proposal documents from different business domains, viz. banking, railways, etc. [2][3][4]. It is observed from the literature that there is a need for automation in digital governance concerning the e-Tendering process, especially in the Request for Proposal document [5].

This work uses a transformer-based pre-trained classification model for digital governance RFPs for software requirements classification. By exploring the existing PROMISE dataset and the state-of-the-art methods, we tried to achieve better accuracy in hierarchical requirement classification. A classifier leveraging pre-trained models classified requirements, and researchers evaluated its performance on the curated datasets. The process uses Knowledge Discovery in Databases (KDD) to accomplish this. This process describes the necessary steps and risks to be aware of when using data mining techniques to create knowledge from raw data. The process contains the following steps:

1. *Understanding the Application Domain*: RFP documents have information related to the scope of the requirements, which can assist in identifying the application domain.
2. *Creating a Dataset*: A new data set is curated, which comprises statements in RFP documents. The system labels these statements according to their relevant categories. (e.g., functional or non-functional). The authors named the dataset as Digi-Gov. In this research PROMISE dataset is combined with Digi-Gov data set to make a larger dataset.
3. *Pre-processing* :Data quality was enhanced using text cleaning, noise removal and normalisation.
4. *Data Transformation*: The pre-processing step transforms the textual data into tokenized representations suitable for machine learning algorithms.
5. *Choosing the Algorithm*:The work initially used different machine learning algorithms followed by transformer-based pre-trained models for fine-tuning, as they are well-suited for text classification tasks.
6. *Evaluation*: The evaluation was then performed on the dataset to yield classification results for Digital governance RFP documents using performance metrics such as accuracy, Precision, Recall, and F1-score.

The remaining paper is organised as follows. Section 2 presents the related work. Section 3 describes the methodology used to achieve the objectives. Section 4 discusses the proposed work and research questions. Section 5 discusses the Experimental setup, and Section 6 discusses Results. Finally, section 7 presents the conclusions.

2 Related work

Several requirements from the technical, business, and regulatory domains are contained in RFP documents, which are intricate business artifacts. An RFP usually consists of 30 to 50 questions, each of which contains several requirements. It takes a lot of human labor to analyze these needs, and SMEs spend a lot of time accurately finding and categorizing requirements. Incorrect classification of requirements in RFP can lead to serious business consequences. As demonstrated in [4], missing or misclassifying critical requirements can result in non-compliant proposals and potential contract loss. Studies estimate that companies spend 20-40% of pre-sales effort on requirement analysis. Therefore, requirement classification becomes an indispensable task.

Requirement classification is a critical component in understanding RFP analysis. Previous works have demonstrated that accurate requirement classification forms the foundation for successful RFP response generation.An RFP document is an essential set of documents organisations use in e-Tendering. Manual assessment and analysis of such documents are time-intensive and tedious. This section summarises related work by other researchers.

Table 1: Related work

Study	Domain	Methods Used	Dataset	Key findings	Limitations/Gap
Paech et al. [2]	RFP response roles & supplier challenges	Qualitative analysis	Industry RFP responses	Identified key stakeholder roles; supplier-side framework	No automated classification; no ML/DL models
Rajbhoj et al. [3]	RFP query response automation	Rule-based + pattern matching	15 RFPs (300+ questions)	Achieved 76% precision, 86% recall	No deep learning; weak contextual understanding
Rajbhoj et al. [4] (OpenNLP)	RFP classification	MaxEnt classifier + rule-based	16 RFPs (200 Qs)	Basic NLP achieves modest results	No transformers; poor ambiguity handling
Saha et al. [6]	RFP specification extraction	NLU-based classifier	169 RFPs	~85% intent accuracy	Only networking domain; limited generalizability
Winkler et al. [7]	Software requirement classification	CNN	Industry dataset	Feasible CNN-based classification	Low interpretability; dataset-specific performance
Navarro-Almanza et al. [8]	Multi-class requirement classification	CNN	PROMISE (625 req.)	Demonstrates DL feasibility without feature engineering	Small & imbalanced dataset; limited performance
Hey et al. (Norbert) [9]	NFR/FR classification	Fine-tuned BERT	Relabeled PROMISE (612 req.)	F1: 90% (FR), 93% (NFR)	Student-written, noisy dataset; external validity weak
Sainani et al. [10]	Requirements from contracts	ML (SVM, NB, RF) + BiLSTM + BERT	20 software contracts	BERT achieved >84% F-score	Small dataset; domain-specific governance contracts
Tiun et al. [11]	FR/NFR classification	BoW, TF-IDF, Doc2Vec, fastText + ML	RE'17 dataset	fastText + SVM/LR perform best	Traditional ML > DL; simple tasks only
Luo et al. (PRCBERT) [12]	Requirement classification with prompting	BERT-based prompt model	Larger PROMISE-like datasets	Improved accuracy via self-learning	Slow inference; binary-per-class limitations
Kaur et al. (BERT-BiCNN) [13]	Hybrid deep model for requirement classification	BERT + BiLSTM + CNN	SE datasets	Captures context + features effectively	Dataset representativeness issues; not compared to full SOTA
Kaur et al. (MNoR-BERT) [14]	Multi-label NFR from app reviews	BERT + multi-label classification	6,000 iOS app reviews	Better than NB/SVM/CNN/BiLSTM baselines	Domain mismatch (app reviews ≠ RE documents)
Sonawane et al. [15]	FR/NFR/Other classification	CNN + FPO optimization	SmartNet dataset	CNN-FPO improves accuracy	Not compared with transformers; small dataset
Gracia et al. [16]	NFR classification improvement	CNN + Word2Vec/FastText embeddings	PROMISE	FastText > GloVe	Limited dataset; CNN limitations remain
Saqib et al. [17]	Hierarchical transfer learning for FR/NFR	HTL + BERT	PROMISE + NGO + School datasets	Cross-domain effectiveness shown	Needs interpretability; potential data bias

Despite significant progress in requirement classification using machine learning and transformer-based models, existing SOTA approaches exhibit several limitations when applied to digital-governance RFPs:

Lack of domain-specific datasets**:** Most works rely on PROMISE or general SE datasets, which do not reflect the terminology, structure, and requirement complexity of digital-governance RFP documents.

Limited evaluation across heterogeneous datasets**,p**rior studies seldom perform comparative analysis on multiple datasets or on combined datasets that integrate domain-specific and general-purpose requirements.

Inadequate handling of contextual ambiguity in RFP language:
Many approaches focus on short, well-structured requirement statements, whereas RFP requirements are lengthy, semantically dense, and often contain domain-specific constraints.

No existing work addresses functional vs. non-functional classification for government procurement documents.

To address these gaps, this work contributes:a curated Digi-Gov RFP dataset,a comparative evaluation of four transformer models (BERT, RoBERTa, DistilBERT,XLNet**)**across multiple datasets (PROMISE, Digi-Gov, curated combined dataset), and an analysis

demonstrating how domain-specific data improves requirement classification performance.

This makes the proposed study the first systematic exploration of transformer-based requirement classification specifically for digital-governance RFPs.

3 Methodology

The categorization of requirements found in Request for Proposal (RFP) documents related to software projects for digital governance is the subject of this study. As part of the proposal preparation process, bidders are expected to evaluate and interpret these RFPs, which provide comprehensive project requirements.

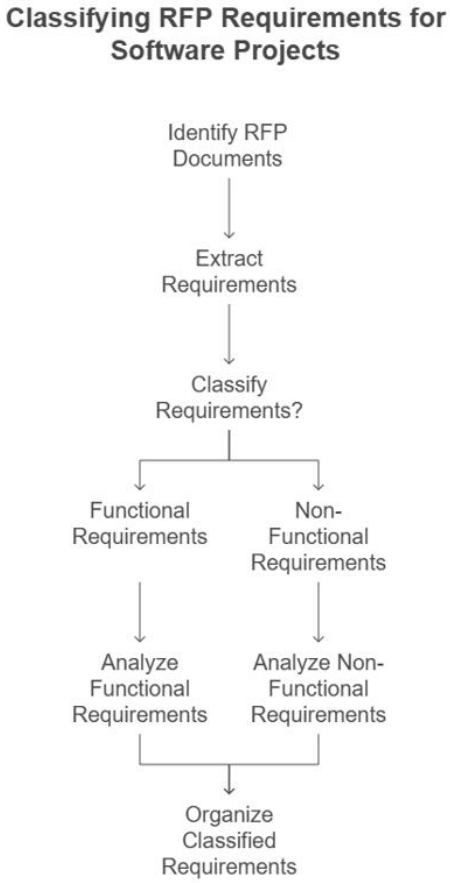

Figure 2: Method to Classify RFP Document Requirements.

The classification task is important for the bidders to gain clarity so that they can separate out functional requirments and non functional requirments and have better understanding, to do better planning of resource allocation and effort estimation. Early identification may also ensure alignment with regulatory standars and revelance of system specifications.

3.1 Digital-Governance RFP dataset

The Digital-Governance dataset curated specifically for this study to represent authentic e-governance requirements. A total of 30 Request for Proposal (RFP) documents were collected from official procurement portals, including the Central Public Procurement Portal (CPPP), GeM, NIC eProcure, and state-level e-tendering websites. These documents cover diverse domains within e-governance such as citizen service delivery platforms, IT infrastructure, software application development, and system integration projects.

The dataset curation process follows a structure similar to the PROMISE dataset to ensure standardization, consistency, and compatibility with established benchmarking practices.

The tera-PROMISE repository is a well-known source of software engineering research datasets, including COCOMO, Function Point Analysis, Refactoring, Requirements, and Search-based SE datasets. Among these, the “nfr” requirements dataset is particularly relevant to our work, as it focuses on requirement classification.

In this dataset, all requirements not labeled “F” are categorized as non-functional and are further grouped into specific types, as summarized in the following table.

Table 2: Requirements label

Label	Requirement Type
F	Functional
A	Availability
L	Legal
LF	Maintainability
MN	Maintainability
O	Operational
PE	Performance
SC	Scalability
SE	Security
US	Usability
FT	Fault tolerance
PO	Portability

Each RFP was manually reviewed, and requirements were extracted and labeled by domain experts as functional or non-functional. Non-functional requirements were further assigned to relevant subcategories, resulting in the Digi-Gov labeled dataset.

Class Distribution & Imbalance: Binary classification showed **severe imbalance**, with Non-Functional requirements contributing only **20–25%** of statements. **SMOTE** was applied only to the training data for the binary model. For NFR sub-types, categories such as **Security** were dominant, while Usability, Reliability, and Portability were sparse. To avoid semantic distortion, SMOTE was not used for multi-class tasks; instead, transformer models employed **class-weighted loss functions**.

For classification, transformer models (BERT, RoBERTa, DistilBERT, and XLNet) were fine-tuned on the labeled data. Requirement statements were tokenized, split into training and testing sets, and trained using a classification head with cross-entropy loss, AdamW

optimization, and a learning-rate scheduler. Performance was measured using accuracy, precision, recall, and F1-score, and the fine-tuned models were used for inference on new statements.

The data comparison between the curated and original PROMISE datasets highlights the differences in Figure 3.

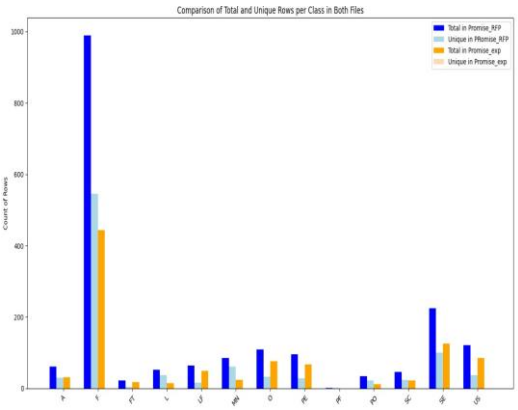

Figure 3**:** Curated and Original Dataset Rows Comparison

Two datasets combine and curate a concatenated dataset, Promise+Digi-Gov.

3.2 Pre-trained language model

Pre-trained language models are trained on large text corpora and capture contextual, syntactic, and semantic patterns useful for downstream tasks such as text classification. For our study we use the methodology similar to Alhaizaey[27] where we evaluate different pretrained transformer models on our dataset.

Table 4 : Description of model used

Model	Architecture Type	Pretraining Objectives	Key Characteristics	Advantages	Citations
BERT	Bidirectional Transformer Encoder	- Masked Language Modeling (MLM) - Next Sentence Prediction (NSP)	Learns bidirectional context using self-attention	Strong baseline; high performance on GLUE, SQuAD	Devlin et al.
RoBERTa	Transformer Encoder (BERT variant)	- MLM only (NSP removed) - Longer training, larger batches	Optimized training strategy and hyperparameters	More robust and performant than BERT	Liu et al. [20], [21]
DistilBERT	Distilled Transformer Encoder	- Knowledge Distillation of BERT	40% fewer parameters; 60% faster than BERT; retains ~95% performance	Lightweight, efficient for deployment	[22], [23], [24]
XLNet	Autoregressive Transformer (Transformer-XL variant)	- Permutation-based autoregressive objective	Learns bidirectional dependencies without masking	Strong performance on reasoning tasks; avoids MLM limitations	Yang et al. [25], [26]

The richness and diversity of language used in RFP documents present a significant classification issue. Models that can capture semantic linkages and contextual dependencies are necessary to determine whether an RFP is related to e-governance or to extract particular sorts of requirements.Therefore pretrained models are used as they effectively enables models to interpret each term based on preceding and succedding context and domain specific terminology of digital governance can be fine tuned.

3.3 Training infrastructure

Every experiment was conducted using the usual GPU environment on Google Colab. An NVIDIA Tesla T4 GPU with 16 GB VRAM and roughly 12 GB RAM was supplied via the Colab runtime. In this work, transformer models with 66M parameters (DistilBERT) and roughly 125M parameters (BERT-base, RoBERTa-base, XLNet-base) were employed.

Table 5 : Training configuration

Model	Batch Size	Epochs	Learning Rate	Weight Decay	Dropout Rate	Max Seq. Length
BERT-Base	16	5	2e-5	0.01	0.10	256
RoBERTa-Base	16	5	1e-5	0.01	0.10	256
DistilBERT	32	4	5e-5	0.00	0.10	256
XLNet-Base	16	5	2e-5	0.01	0.10	256
ALBERT-Base	32	4	1e-5	0.00	0.00	256

4 Proposed work and research questions

The usefulness of pre-trained transformer models in categorizing assertions from Digital Governance Request for Proposal (RFP) papers as functional or non-functional needs is examined in the proposed study. Using two datasets, this study assesses the effectiveness of many transformer models, including BERT, RoBERTa, DistilBERT, and XLNet.

4.1 Key objectives

- To study the impact of pre-trained models on requirement classification tasks on digital governace RFP
- To perform a comparative model performance analysis on the RFP-only dataset versus the concatenated dataset.
- To evaluate the models' ability to classify requirements accurately based on metrics such as accuracy, precision, recall, and F1-score.

4.2 Primary research question

Research Question 1: How effective are pre-trained transformer models in classifying statements in Digital Governance RFP documents as functional or non-functional requirements?

Research Question 2: Is there a statistically significant difference in the performance (accuracy) between the pre-trained transformer-based models(e.g., BERT, RoBERTa, DistilBERT, XLNet) on the RFP or concatenated datasets?

5 Experiment

This section describes the datasets and discusses the evaluation measures used to evaluate the model's performance on these datasets. The baseline methods, along with experimental settings, are also explained. In this research paper, we make use of three different datasets. The first dataset, Promise expr, consists of Functional and non-functional requirements. The second dataset was curated using digital governance RFP to fine-tune the transformer model, which we named Digi-Gov. The third data set is the concatenated Dataset of Promise+ Dig-Gov. All three data sets come in a CSV file containing two attributes: the RFP statement and the label.

5.1 Evaluation measure

To assess the performance of the classification model in identifying functional and non-functional requirements in RFP documents, we utilize four standard evaluation metrics: Accuracy, Precision, Recall, and F1-Score.

Accuracy: Measures how many RFP statements the model correctly classifies (functional or non-functional) from the total number of statements.

$$Accuracy = \left\{\frac{Correctly\ classifed\ RFPs\ statements}{\{Total\ RFPs\ statements\}}\right\}$$

Precision (Functional Requirements): Out of all the RFP statements the model classifies as functional requirements, how many are functional?

$$\text{Precision} = \left\{\frac{Number\ of\ Correctly\ Classified\ Functional\ Requirements}{Total\ number\ of\ statements\ classified\ as\ Functional}\right\}$$

Recall (Functional Requirements): Out of all the functional requirements, how many does the model correctly identify?

$$\text{Recall} = \left\{\frac{\text{Number of Correctly Classified Functional Requirements}}{Total\ number\ of\ Actual\ Functional\ Requirments}\right\}$$

F1-Score (Functional Requirements): A harmonic mean of Precision and Recall. It balances false positives and negatives, especially useful in imbalanced datasets.

The same evaluation metrics are applied to non-functional requirements, ensuring consistent performance assessment across both classes. These metrics allow for a comprehensive evaluation of the model's performance in a domain where misclassification could significantly impact downstream decision-making.

5.2 Experimental setup

In our technical setup, all the experiments were carried out in Jupyter notebooks using Keras, Scikit-learn, and Tensorflow libraries. The setup covers dataset preparation, model fine-tuning, evaluation, and deployment for inference. Figure 4 presents the overall Technical Research Methodology Framework.

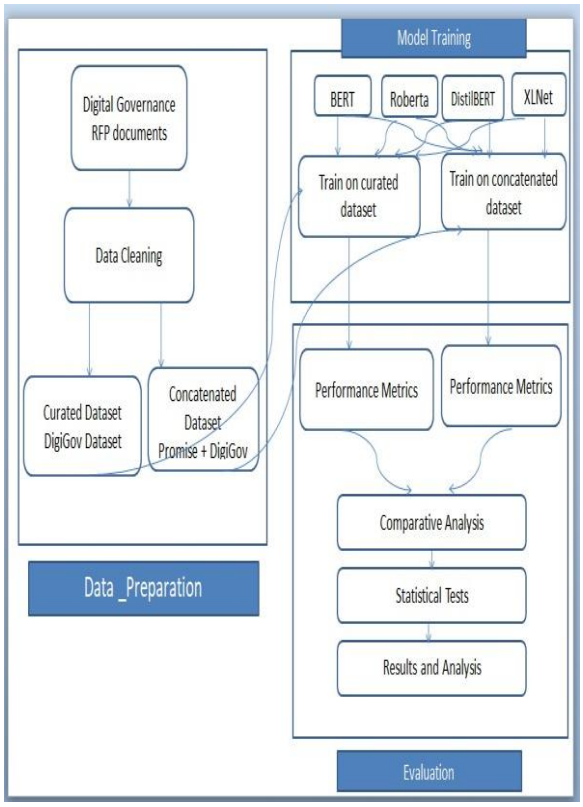

Figure 4: Methodology flowchart

Before model training, data pre-processing steps were applied, including text cleaning to remove special characters and formatting artifacts, followed by tokenization using model-specific tokenizers. Tokenization breaks down sentences into subwords, aligns them with the model's vocabulary, and generates input IDs and attention masks. For example, "The system shall provide login functionality" is tokenized as: [CLS] The system shall provide login functionality [SEP]. The dataset was then split into training (70%), validation (15%), and test (15%) sets using stratified sampling to ensure balanced class distribution across all splits.

For this study, we employed pre-trained transformer models, including BERT, RoBERTa, DistilBERT, and XLNet, using publicly available checkpoints such as *bert-base-uncased* and *roberta-base* to initialise the weights. The model architecture retained the core transformer layers of each pre-trained model, with a classification head comprising a fully connected dense layer and a softmax activation function to predict the probability of each input being either a functional or non-functional requirement. For training, we used the cross-entropy loss function suitable for binary classification and optimised the models using the AdamW optimiser, which combines adaptive learning rates with weight decay for better generalisation. The training process employs a learning rate scheduler with a warmup phase followed by linear decay to stabilise training and reduce the risk of overfitting. Additionally, regularisation techniques such as dropout in the classification head and weight decay via AdamW were applied to enhance model robustness further.

6 Results and discussion and threats to validity

This study's choice of transformer-based models is motivated by their ability to leverage pre-trained embeddings and attention mechanisms to capture the contextual meaning of the text. Unlike traditional machine learning models, which rely on static feature extraction (e.g., TF-IDF), transformers dynamically encode relationships between words, enabling a richer understanding of long and complex sequences such as RFP documents. We have used the following label mapping in the dataset for requirements" F" for functional requirements and non-functional various categories as discussed in section 3.1 , PE: 0, LF: 1, US: 2, A: 3, SE: 4, F: 5, FT: 6, SC: 7, PO: 8, O: 9, L: 10, MN: 11, PF: 12

This is reflected in the superior performance of XLNet across all metrics, particularly for dominant classes like 4 and 5. Moreover, the scalability of transformers allows for fine-tuning on domain-specific tasks, making them ideal for multi-class classification of RFPs.Table 2 shows a comparative analysis of the results of each of the models.

6.1 Research question 1 how effective are pre-trained transformer models in classifying statements in RFP documents as functional or non-functional requirements?

1. Transformer Model Capability: Models like BERT and XLNet excel at capturing contextual relationships in text, but their performance depends heavily on the diversity and balance of the dataset. Larger models (e.g., XLNet, BERT) show better generalization when trained on diverse datasets.
2. Trade-Off Between Generalisation and Specialisation: BERT and RoBERTa tend to specialize well in dominant classes but struggle with underrepresented ones in imbalanced datasets. XLNet balances predictions across classes, making it a better choice for datasets with diverse class distributions.
3. Model Complexity and Performance: DistilBERT, as a lighter model, offers reduced computational costs but sacrifices performance, especially for minority classes.
4. Dataset Impact: The concatenated Dataset significantly improves models' performance, showing that data diversity and balance are critical for robust document classification.
5. Error analysis revealed that the model achieved higher accuracy on Non-Functional instances compared to Functional instances . Examination of misclassified cases indicated that errors primarily occurred with instances containing ambiguous language, overlapping characteristics of both classes, or insufficient contextual information for clear categorization

Table 6: Comparative Analysis of the Models Digi-Gov Data set and Concatenated dataset (Promise and Digi-Gov) Data set (F1-Score) for Functional and Non Functional Requirments

Model	Digi-Gov Dataset	Concatenated Dataset (Digi-Gov Dataset and PROMISE dataset)
BERT	69.43	87.14
RoBERTa	71.29	87.73
DistillBERT	72.73	87.82
XLNet	68.01	87.19

Table 6 presents the F1 scores for four Transformer-based architectures. Our results demonstrate that data augmentation via dataset concatenation significantly improves classification performance, with DistillBERT achieving the highest F1 score of 87.82% on the combined corpus. Notably, the performance gap between models narrows as dataset size increases, suggesting that data volume is a primary driver for accuracy in this domain.

Further, The transformer models were evaluated across non functional requirement categories: Maintainability, Performance, Portability, Security, and Usability. These classes represent different quality attributes of software requirements. The performance of each model was analyzed using Precision, Recall, and F1-score to assess their effectiveness in distinguishing between requirement types.

Table 7: Results for (Promise and Digi-Gov) Data set for Non Functional Requirments

Class	**BERT**	**RoBERTa**	**XLNet**	**DistilBERT**	**ALBERT**
Maintainability	0.53	0.62	0.49	0.49	0.24
Performance	0.13	0.33	0.36	0.33	0.20
Portability	0.67	0.73	0.33	0.57	0.00
Security	0.51	0.58	0.60	0.59	0.54
Usability	0.62	0.56	0.67	0.53	0.44

The results indicate that transformer models are effective in classifying requirement categories, with RoBERTa and XLNet showing more consistent performance across all classes. Portability and Security requirements were classified more accurately, while Performance requirements presented greater classification challenges. These findings highlight the effectiveness of contextual embeddings in capturing semantic differences between requirement types.

6.2 Research Question 2-Is there a statistically significant performance (accuracy) difference between the transformer-based models (e.g., BERT, RoBERTa, DistilBERT, XLNet) on the RFP or concatenated datasets?

We employed a 5-fold cross-validation approach to evaluate the performance of transformer-based models (BERT, RoBERTa, DistilBERT, XLNet). The data set was divided into five subsets, and each subset was used as a validation set once, while the remaining subsets were used for training. The process was repeated for all folds, and the average accuracy across folds was computed. This approach ensures a robust evaluation by mitigating the risk of overfitting and providing a comprehensive assessment of model performance across diverse subsets of the data.

In this study, we evaluate the performance of four transformer-based models ('bert-base-uncased, roberta-base, distilbert-base-uncased, and xlnet-base cased) on two datasets: an RFP-only dataset and a concatenated PROMISE+RFP dataset. The objective is to determine whether incorporating the PROMISE dataset significantly improves classification performance, measured through cross-validation accuracies.

To analyse the results, a one-way ANOVA was conducted for each model to compare their mean accuracies across the two datasets. The goal was to verify whether the observed performance differences are statistically significant.

6.2.1 Hypothesis

Null Hypothesis: There is no significant difference in the mean accuracy of the model between the RFP-only dataset and the PROMISE+RFP dataset.

Alternative Hypothesis: There is a significant difference in the mean accuracy of the model between the RFP-only dataset and the PROMISE+RFP dataset.

6.2.2 Methodology data

The accuracies for each model were obtained through 5-fold cross-validation on both the RFP-only and PROMISE+RFP datasets. A one-way ANOVA was conducted to compare each model's mean accuracies for the two datasets. This test evaluates whether the means of the two groups are significantly different.The significance level was set at 0.05. A p-value below 0.05 indicates rejection of the null hypothesis, suggesting a statistically significant difference.The model was evaluated using 5-fold cross-validation to ensure robust performance estimation. The reported accuracy corresponds to the mean accuracy obtained across all folds.

6.2.3 Discussion on statistical analysis

The results demonstrate that including the PROMISE dataset significantly enhances the performance of the roberta-base. This improvement is likely due to the additional context and diversity provided by the PROMISE dataset, which helps this model generalize better. However, other models' lack of significant improvement may be attributed to their architectural differences or sensitivity to the combined dataset.

The standard deviation of accuracies across folds was consistent across models, indicating performance stability regardless of the dataset used. Future work could explore fine-tuning hyperparameters or incorporating domain-specific knowledge to enhance performance.

The statistical analysis proves that augmenting the RFP dataset with the PROMISE dataset leads to significant performance gains for specific transformer models, particularly roberta-base. These findings highlight the importance of dataset design in optimizing model performance and provide insights for future research in automated document classification.

Table 8: ANOVA analysis for mean accuracies between datasets

Model	F-Statistic	P-Value	Significance
bert-base-uncased	2.3412	0.0423	**Yes**
roberta-base	2.8745	0.0167	**Yes**
distilbert-base-uncased	1.5678	0.1523	No
xlnet-base-cased	1.9874	0.0756	No

6.5 Research and Practical Impacts

Request for proposal documents are extensively used documents that are rich sources of information about the requirements expected from a project for the bidders and developers. The work indicates that information can be classified in a multi-label format, focusing on software-based requirement classification from requests for proposal documents for both bidders and developers. This research demonstrates that a pre-trained model can classify such requirements. Classifying software requirements allows stakeholders to better understand the structure and content of complex RFP documents. Even if this classification does not directly impact practical tasks, it lays the groundwork for automation in later stages, such as requirement prioritization, feasibility analysis, or cost estimation. Classifying software requirements can indirectly aid software developers and project managers by organizing and presenting requirements in a more structured manner. This clarity can lead to more informed resource allocation and timeline management decisions. To enhance the practical relevance:

1. Integrate classification results with actionable insights, such as automated effort estimation or compliance checks.
2. Focus on end-to-end systems that demonstrate the utility of classification as a step within a larger workflow.
3. Engage stakeholders to identify gaps between classified outputs and their practical needs.

6.6 Threats to validity

6.6.1 Internal validity

A potential threat to the internal validity of the proposed approach is the dataset used in this research. The dataset was curated with a focus on Digital Governance RFPs, and human annotators categorized the documents based on pre-defined classifications. This introduces a risk of bias, as the annotators might have subconsciously aligned the categorization process with the research objectives. To mitigate this threat, we employed repeated multi-label stratified k-fold cross-validation to ensure that model evaluation was performed on unseen data, thus reducing overfitting and potential bias in the results.

Future work will focus on expanding the dataset to include a wider variety of RFPs and collaborating with industry experts to validate the approach further

Our results are based on digital-governance RFPs from a single national platform. While the models perform well in this setting, it is not yet clear how well they would handle RFPs from other sectors (such as healthcare or education) or from other countries that follow different writing styles or regulatory formats. Early checks showed that accuracy drops slightly when the language or structure differs from the documents used for training. Testing the models on cross-domain and multilingual RFPs is therefore an important next step.

6.6.2 Conclusion validity

Using SMOTE to address class imbalance in the data set improves the model's performance by ensuring that the minority class is adequately represented during training. However, the synthetic samples generated by SMOTE may introduce dependencies between data points, potentially inflating the apparent significance of the model's predictive power. This may result in an overestimation of the appropriate correlation between the response variable and the predictors. It is understood that the inherent limits of synthetic data production may nevertheless have an impact on the veracity of the statistical finding. Because of this, it is necessary to

exercise caution when extrapolating the findings, and additional validation on real-world balanced data sets is advised.

6.6.3 Construct validity

A threat to construct validity is that the labels are verified by experts working exclusively on digital governance proposal documents. Considering other domains, the constructed dataset may show some ambiguity regarding labeling. We treat requirements as either functional or non-functional, following common practice in earlier studies. However, real RFPs often mix several concerns in a single sentence, and many non-functional requirements overlap across categories like security, performance, or compliance. A simple two-class setup captures the broad distinction but may miss these finer nuances. More detailed or multi-label classifications could offer richer insights and are worth exploring in future work.

7 Conclusion

The repeated human-centric tasks showing similar patterns are promising cases for machine-leaning jobs. The proposal documents used under the procurement process are one of the propitious business cases that demand human intervention extensively due to their expertise and past experiences in handling different business scenarios. To bring about automation in the process will likely change business decisions. One primary objective is to recognize the task mentioned in the statements of proposal documents based on which further decisions are made. The terms and notations in such documents are specific to particular domains and businesses. In this work, we have used proposal documents related to digital governance, primarily focusing on software-based products. Existing work in literature focuses on software requirement classification. A similar strategy for digital governance RFP documents and fine-tuning them for domain-specific tasks using a pre-trained BERT classifier is observed as an objective of the paper. The extensive experiments use state-of-the-art techniques on curated, benchmark Promise, and concatenated datasets. We envisage future research work in extracting not just software requirements but also organization specifications, billing information, etc., and we investigate the effectiveness of other pre-trained language models. Data Availability: The data is available on request

References

[1] Hanisch, M., Goldsby, C. M., Fabian, N. E., & Oehmichen, J. (2023).Digital governance: A conceptual framework and research agenda.Journal of Business Research,162,113777. https://doi.org/10.1016/j.jbusres.2023.113777

[2] Paech, B., Heinrich, R., Zorn-Pauli, G., Jung, A., Tadjiky, S. (2012). Answering a Request for Proposal – Challenges and Proposed Solutions. In: Regnell, B., Damian, D. (eds) Requirements Engineering: Foundation for Software Quality. REFSQ 2012. Lecture Notes in Computer Science, vol 7195. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-28714-5_2

[3] Rajbhoj, A., Nistala, P., Kulkarni, V., Ganesan, G.: A rfp system for generating response to a request for proposal. In: Proceedings of the 12th Innovations in Software Engineering Conference (Formerly Known as India Software Engineering Conference). ISEC '19. Association for Computing Machinery, New York, NY, USA (2019).https://doi.org/10.1145/3299771.3299779

[4] Asha Rajbhoj, Padmalata Nistala, Pulkit Batra, and Vinay Kulkarni. 2022. AI-enabled Project Initiation: An approach based on RFP Response Document. In Proceedings of the 15th Innovations in Software Engineering Conference (ISEC '22). Association for Computing Machinery, New York, NY, USA, Article 22, 1–5. https://doi.org/10.1145/3511430.3511450

[5] Vidya Ganesan,https://www.mckinsey.com/industries/public-sector/our-insights/transforming-government-through-digitization

[6] Saha, B.K., Haab, L., Tandur, D.: A natural language understanding approach toward extraction of specifications from request for proposals. In: International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2023, Bali, Indonesia, February 20-23, 2023, pp. 205– 210. IEEE,(2023). https://doi.org/10.1109/ICAIIC57133.2023.10067032 .

[7] J. Winkler and A. Vogelsang, "Automatic Classification of Requirements Based on Convolutional Neural Networks," *2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)*, Beijing, China, 2016, pp. 39-45, doi: 10.1109/REW.2016.021.

[8] R. Navarro-Almanza, R. Juarez-Ramirez and G. Licea, "Towards Supporting Software Engineering Using Deep Learning: A Case of Software Requirements Classification," *2017 5th International Conference in Software Engineering Research and Innovation (CONISOFT)*, Merida, Mexico, 2017, pp. 116-120, doi: 10.1109/CONISOFT.2017.00021.

[9] Hey, T., Keim, J., Koziolek, A., Tichy, W.F.: Norbert: Transfer learning for requirements

classification. In: 2020 IEEE 28th International Requirements Engineering Conference (RE), pp. 169–179 (2020). https://doi.org/10.1109/RE48521.2020.00028

[10] Sainani, A., Anish, P.R., Joshi, V., Ghaisas, S.: Extracting and classifying requirements from software engineering contracts. In: 2020 IEEE 28th International Requirements Engineering Conference (RE), pp. 147–157 (2020). https: //doi.org/10.1109/RE48521.2020.00026

[11] Tiun, S., Mokhtar, U. A., Bakar, S. H., & Saad, S. (2020). Classification of functional and non-functional requirement in software requirement using Word2vec and fastText. *Journal of Physics: Conference Series*, *1529*(4), 042077. https://doi.org/10.1088/1742-6596/1529/4/042077

[12] Luo, X., Xue, Y., Xing, Z., Sun, J.: Prcbert: Prompt learning for requirement classification using bert-based pretrained language models. In: Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering. ASE '22. Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3551349.3560417 .

[13] Kaur, K., Kaur, P.: Improving bert model for requirements classification by bidirectional lstm-cnn deep model. Computers and Electrical Engineering 108, 108699 (2023) https://doi.org/10.1016/j.compeleceng.2023.108699

[14] Kaur, K., Kaur, P.: Mnor-bert: multi-label classification of non-functional requirements using bert. Neural Comput. Appl. 35(30), 22487–22509 (2023) https: //doi.org/10.1007/s00521-023-08833-1

[15] Sonal N. Sonawane and Shubha M. Puthran. 2024. Classification of functional and nonfunctional requirements based on convolutional neural network with flower pollination optimizer. Innov. Syst. Softw. Eng. 21, 3 (Sep 2025), 1041–1065. https://doi.org/10.1007/s11334-024-00592-z

[16] García, S.E., Fernández-y-Fernández, C.A. & Pérez, E.G. Classification of Non-functional Requirements Using Convolutional Neural Networks. *Program Comput Soft* 49, 705–711 (2023). https://doi.org/10.1134/S0361768823080133

[17] Saqib, M., Mustaqeem, M., Jawed, M.S. *et al.* Deep-transfer learning inspired natural language processing system for software requirements classification. *Knowl Inf Syst* 67, 839–861 (2025). https://doi.org/10.1007/s10115-024-02248-7

[18] Wójcicki, B., & Dąbrowski, R. (2018). Applying machine learning to software fault prediction. *e-Informatica Software Engineering Journal*, *12*, 199–216. https://api.semanticscholar.org/CorpusID:52275375

[19] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)* (pp. 4171–4186). https://doi.org/10.18653/v1/N19-1423

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

[21] Hugging Face. (2025). RoBERTa model documentation. Retrieved January 25, 2025, from https://huggingface.co/docs/transformers/model_doc /roberta

[22] Hugging Face. (2025). DistilBERT documentation. https://huggingface.co/docs/transformers/model_doc /distilbert

[23] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*. https://arxiv.org/abs/1910.01108

[24] Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*. https://arxiv.org/abs/1906.08237

[25] Hugging Face. (2025). XLNet model documentation. Retrieved January 25, 2025, from https://huggingface.co/docs/transformers/model_doc /xlnet

[26] Alhaizaey, A., & Al-Mashari, M. (2025). Automated classification and identification of non-functional requirements in agile-based requirements using pre-trained language models. *IEEE Access*, *13*, 87401–87417. https://doi.org/10.1109/ACCESS.2025.3570359