NLP-based Page Classification for Efficient LLM Extraction from Brazilian Public Tender Documents

Pedro Campos[1], **Ivo de Medeiros**[2], **and Adailton de Araújo**[1]
[1] Universidade Federal de Goiás (UFG), Goiânia, Brazil
[2] CEIA-UFG, Goiânia, Brazil
campos23@discente.ufg.br ivopdm@gmail.com adailton@ufg.br

Abstract

Extracting structured information from lengthy documents using Large Language Models (LLMs) is computationally expensive and prone to accuracy degradation as input size increases. We present a two-stage pipeline for extracting products from Brazilian tender documents (*editais de licitação*), combining NLP-based page classification with LLM extraction. We construct a novel dataset of 11,190 annotated pages from 350 documents across five product domains. Our experiments compare transformer-based classifiers (BERTimbau, DistilBERT) with classical machine learning approaches using engineered features. Results show that XGBoost with domain-specific features achieves 97.75% F1-score, outperforming fine-tuned BERT models by over 4 percentage points. The complete pipeline reduces LLM input tokens by 64–88% while maintaining extraction completeness, enabling cost-effective document processing at scale.

1 Introduction

Large Language Models have demonstrated remarkable capabilities in information extraction tasks (Brown et al., 2020). However, processing lengthy documents remains costly, as larger inputs increase API expenses and may degrade extraction accuracy. Brazilian public tender documents (*editais de licitação*) exemplify this challenge: they average nearly 50 pages and contain extensive legal and administrative content, while the target information–product with items, quantities, and prices–occupies only a small fraction of the total document.

In this work, we propose a two-stage approach that leverages NLP classification techniques to identify relevant pages before extraction based on LLM. Our contributions include: (1) a novel annotated dataset of procurement documents, (2) comprehensive evaluation of transformer (Vaswani et al., 2017) and classical ML approaches for page classification, and (3) an end-to-end pipeline achieving substantial token reduction while maintaining extraction quality.

2 Methodology

This section describes our methodological approach, including the problem formulation, the construction and annotation of the dataset, the classification and extraction model architectures, and the evaluation protocol.

2.1 Problem Statement

Extracting structured information from Brazilian public tender documents presents significant challenges for Large Language Models. These documents are lengthy, averaging 49.72 pages (median: 46) with a mean of 38,982 tokens per document (standard deviation: 34,883). The largest documents exceed 165,000 tokens, which, although within the context window of recent LLMs such as Gemini, incurs substantial computational cost and increased API expenses when processed in full.

Directly processing entire documents with LLMs is both computationally expensive and prone to accuracy degradation as input length increases. Furthermore, relevant information, specifically product tables that contain items, quantities, and prices, typically occupies only a small fraction of the total document. Our analysis reveals that only 16.9% of pages (1,887 out of 11,138) contain product tables, while the remaining 83.1% consist of legal clauses, administrative procedures, and other content not relevant for our application.

We propose a two-stage pipeline that leverages NLP classification techniques to filter relevant pages prior to LLM extraction, thus reducing input size by 64–88% while maintaining extraction quality.

Proceedings of the 17th International Conference on Computational Processing of Portuguese (PROPOR 2026)
Vol. 1, pages 621–627

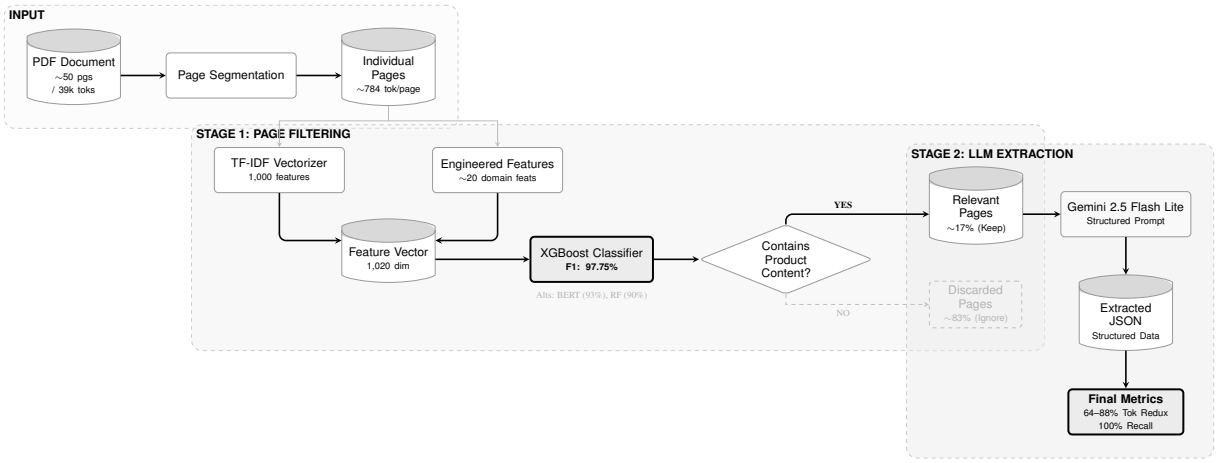

Figure 1: Refined pipeline architecture (Grayscale). Stage 1 utilizes XGBoost to filter pages based on engineered features, significantly reducing the token load for the Stage 2 LLM extraction.

2.2 Dataset

This subsection details the data collection process, corpus composition, and the hybrid annotation strategy employed to construct our novel dataset of Brazilian public procurement documents.

2.2.1 Data Collection and Composition

Due to the absence of publicly available datasets for this domain, we constructed a novel corpus comprising 350 Brazilian public procurement documents totaling 11,190 pages. To ensure generalization across different product domains and avoid classifier bias toward specific vocabulary, we stratified the collection into five product families: hospital medications, audiovisual and musical equipment, dental prosthetics and supplies, and information technology products.

The documents exhibit substantial heterogeneity in structure and length. Token statistics per page show a mean of 784 tokens (median: 727, σ: 368.36), with pages containing tables averaging 1,023 tokens compared to 735 tokens for non-table pages. Document-level statistics reveal high variance, with page counts ranging from 1 to 253 pages per document.

2.2.2 Annotation Process

We employed a hybrid annotation strategy combining manual labeling with LLM-assisted classification. The annotation pipeline consisted of three stages: document segmentation, manual annotation, and LLM-assisted expansion.

Document Segmentation. Each PDF document was programmatically segmented into individual pages, with text extracted and stored separately. This page-level granularity enables precise classification and subsequent filtering for the extraction pipeline.

Manual Annotation. Domain experts manually annotated 6,441 pages using binary labels: `1` (contains product/item tables) and `0` (no relevant tabular information). This manually annotated subset served as both training data and a validation benchmark for the LLM-assisted annotation.

LLM-Assisted Annotation. To scale the annotation process, we developed a comprehensive few-shot prompt for Gemini 2.5 Flash Lite.[1] Before applying the LLM to unannotated pages, we validated its reliability by classifying the same 6,441 manually annotated pages, using our manual labels as ground truth.

The LLM achieved 99.84% accuracy, 99.47% precision, 100% recall, and 99.73% F1-score (confusion matrix: TN=4,554, FP=10, FN=0, TP=1,877). These high metrics can be attributed to the careful prompt engineering: (1) the few-shot examples were carefully selected to cover diverse edge cases across different product domains and table formats; (2) we designed an algorithmic decision tree within the prompt to guide the model's reasoning process; and (3) we explicitly documented special cases and boundary conditions encountered during manual annotation. This comprehensive prompt design enabled reliable automated annotation of the remaining corpus pages.

The final dataset distribution comprises 4,243 pages with product tables (38.3%) and 6,829 pages

[1]The complete prompt is available at https://hypernotepad.com/n/134080fc06d6ca97

without (61.7%). We partitioned the data into 80% training (8,857 pages) and 20% test (2,215 pages) sets with stratified sampling.

2.3 Model Architecture

Our pipeline consists of two sequential blocks: a classification module that identifies relevant pages, followed by an extraction module that processes only the filtered content.

2.3.1 Classification Block

We evaluated two complementary approaches for page classification: transformer-based models and classical machine learning with engineered features.

Transformer Models. We fine-tuned two Portuguese BERT (Devlin et al., 2019) variants:

- **BERTimbau** (Souza et al., 2020): A BERT-base model (110M parameters) pre-trained on a large Brazilian Portuguese corpus.
- **DistilBERT-Portuguese** (Sanh et al., 2019): A distilled version (66M parameters) offering reduced computational requirements.

Both models were fine-tuned for binary sequence classification with the following hyperparameters: maximum sequence length of 512 tokens, learning rate of 2×10^{-5}, batch size of 16 (effective 32 with gradient accumulation), and FP16 mixed precision training.

We investigated three tokenization strategies to handle documents exceeding 512 tokens:

1. **Baseline**: Standard truncation retaining only the first 512 tokens.
2. **Stratified Sampling**: For texts exceeding 512 tokens, we divide the content into three equal chunks (beginning, middle, end) and randomly sample approximately 170 contiguous tokens from each, preserving information from all document regions.
3. **Stratified + Cleaning**: Combines stratified sampling with text preprocessing including Unicode normalization (NFKC), removal of visual noise patterns (separator lines, repeated characters), URL/email replacement with placeholder tokens, and whitespace normalization.

Classical Machine Learning. We developed a feature engineering approach combining TF-IDF representations (Salton and Buckley, 1988) with domain-specific handcrafted features, totaling approximately 1,020 input dimensions.

TF-IDF Features: We applied TfidfVectorizer with 1,000 maximum features, unigrams and bigrams, and sublinear term frequency scaling. Text preprocessing included lowercase conversion, removal of common procurement stopwords (*edital*, *prefeitura*, *município*), while preserving structural punctuation.

Engineered Features (approximately 20 dimensions): These capture structural and content patterns indicative of tabular data:

- **Structural**: line count, short line ratio, words per line, pipe character frequency, multiple consecutive spaces (column alignment indicator), whitespace density.
- **Numerical**: number density, monetary pattern frequency (R$, *reais*), percentage occurrences.
- **Domain Keywords**: frequency of terms such as *item*, *quantidade*, *unidade*, *valor*, *preço*, *total*, *lote*.

We trained three classifiers on the combined feature set: XGBoost (Chen and Guestrin, 2016) with scale_pos_weight for class imbalance, Random Forest (Breiman, 2001) with balanced class weights, and SVM (Cortes and Vapnik, 1995) with RBF kernel.

Feature Importance Analysis. To understand which features contribute most to classification performance, we analyzed feature importance scores from the tree-based models. For Random Forest, we used the Gini Importance (Mean Decrease in Impurity), which measures how much each feature contributes to reducing impurity across tree splits. For XGBoost, we used the Gain metric, representing the average information gain from splits using each feature. Table 1 presents the top-ranked features for each model.

2.3.2 Extraction Block

For the extraction stage, we employed Gemini 2.5 Flash Lite (Team et al., 2023) with a structured prompt designed to extract product information (item number, description, quantity, unit, unit price, total price) from the filtered pages. The output

XGBoost		Random Forest	
Feature	**Gain**	**Feature**	**Imp.**
Number Density	0.108	Uppercase Count	0.089
da	0.100	*da*	0.042
Uppercase Count	0.082	*será*	0.027
pública	0.019	*do*	0.025
sp	0.016	*no*	0.023
administrativo	0.015	*dos*	0.022
und	0.015	*100*	0.021
de preços	0.014	*lei*	0.020

Table 1: Top 8 features by importance. Non-italicized features are manually engineered; *italicized* terms are from TF-IDF vectors.

is formatted as JSON for downstream processing. The extraction operates on the concatenated text of pages classified as containing product.

2.4 Evaluation

2.4.1 Classification Metrics

We evaluate classification performance using standard metrics: Accuracy, Precision, Recall, and F1-Score. Given the class imbalance and the application context where missing relevant pages (false negatives) has higher cost than including irrelevant pages (false positives), we prioritize models with high recall while maintaining acceptable precision.

2.4.2 End-to-End Evaluation

For the complete pipeline, we measure:

- **Token Reduction**: Percentage decrease in tokens sent to the LLM compared to processing the full document.
- **Extraction Recall**: Percentage of ground-truth items successfully extracted, measured against manually verified item counts. We report this metric for both the baseline (full document processing) and the proposed pipeline to enable direct comparison.
- **Processing Time**: Total pipeline execution time including classification and extraction.

2.4.3 Experimental Setup

All experiments were conducted on a workstation equipped with an NVIDIA RTX 3090 GPU (24GB VRAM), AMD Ryzen 9 5900X processor (12 cores, 24 threads), and 64GB RAM, running Ubuntu Server. Transformer fine-tuning utilized the GPU for training and inference, while classical ML models (XGBoost, Random Forest, SVM) were trained on CPU.

Model	Strategy	Acc	Prec	Rec	F1
BERTimbau	Baseline	94.72	94.42	91.64	93.01
	Stratified	94.94	92.80	94.11	93.45
	Strat.+Clean	**95.12**	**94.37**	92.82	**93.58**
DistilBERT	Baseline	94.18	93.48	91.17	92.31
	Stratified	94.67	93.77	92.23	92.99
	Strat.+Clean	94.49	90.48	**93.99**	92.77

Table 2: Transformer classification results (%). Best values per metric in bold. Strat.+Clean denotes Stratified Sampling with text cleaning.

Model	Acc	Prec	Rec	F1
XGBoost	**99.24**	**97.37**	**98.14**	**97.75**
Random Forest	96.77	88.41	93.10	90.70
SVM (RBF)	96.68	86.68	94.96	90.63

Table 3: Classical ML classification results (%) using TF-IDF + engineered features.

3 Results

3.1 Classification Performance

3.1.1 Transformer Models

Table 2 presents the performance of transformer-based classifiers across tokenization strategies.

BERTimbau with Stratified Sampling and Cleaning achieves the highest F1-score (93.58%), providing the best balance between precision (94.37%) and recall (92.82%). The stratified sampling strategy consistently improves recall across both models, with BERTimbau showing a 2.47 percentage point improvement over baseline (91.64% $\rightarrow$ 94.11%). However, this comes at the cost of slightly reduced precision.

Text cleaning acts as a regularizer for larger models: BERTimbau benefits from cleaning (F1: 93.45% $\rightarrow$ 93.58%), while DistilBERT shows degraded precision (93.77% $\rightarrow$ 90.48%), suggesting smaller models rely more on contextual cues removed during preprocessing.

3.1.2 Classical Machine Learning

Table 3 shows the performance of classical ML approaches on the table detection task.

XGBoost substantially outperforms all other approaches, achieving 97.75% F1-score with only 17 total errors (FP=10, FN=7) on 2,228 test pages. Feature importance analysis reveals that engineered features—particularly number density, keyword indicators (*item*, *quantidade*), and separator patterns—dominate the top-ranked features, validating the effectiveness of domain-specific feature engineering for structural classification tasks.

Metric	Baseline	Pipeline
Pages processed	63	17
Input tokens	30,271	7,741
Token reduction	—	74.4%
Ground-truth items	409	
Items extracted	409	409
Extraction recall	100%	100%
Processing time (s)	92.48	107.69

Table 4: Pipeline evaluation on a sample document. Baseline processes the full document; Pipeline uses XGBoost classification. Extraction recall is measured against manually verified item counts.

Metric	Baseline	Pipeline
Total pages	57	57
Pages processed	57	21
Chunks	12	5
Input tokens	32,264	14,073
Token reduction	—	56.4%
Ground-truth items	598	
Items extracted	598	598
Extraction recall	100%	100%
Processing time (s)	127.34	124.92

Table 5: Chunked extraction evaluation on a high-item-count document (598 items). Both approaches achieve 100% extraction recall, but the pipeline reduces token consumption by 56.4%.

3.2 End-to-End Pipeline Evaluation

We evaluated the complete pipeline on 10 procurement documents containing 26–60 items each. Table 4 presents a representative example.

Across all test documents, the pipeline achieves token reductions ranging from 64% to 88% while maintaining 100% extraction recall against manually verified ground truth. The slight increase in total processing time is attributable to the additional classification step; however, the reduced LLM input size yields substantial cost savings for API-based deployments and enables processing of documents that would otherwise exceed context window limits.

3.2.1 High-Item-Count Documents

Documents with more than 300 items present challenges for single-pass extraction due to output token limitations and JSON formatting complexity. In such cases, the baseline approach fails to produce valid structured outputs. To address this limitation, we implemented a chunked extraction strategy that processes filtered pages in batches of 5 pages per LLM call, aggregating results across chunks.

Table 5 presents results for a high-item-count document processed with chunking.

The chunked strategy successfully extracts all 598 items in both approaches, demonstrating that the XGBoost filtering maintains extraction completeness even when combined with chunked processing. Notably, the pipeline approach requires fewer chunks (5 vs 12) due to pre-filtering, resulting in comparable processing time despite the additional classification overhead.

4 Discussion

Our results demonstrate that classical machine learning with domain-specific feature engineering outperforms transformer models for structural classification tasks in this domain. XGBoost achieves 97.75% F1 compared to 93.58% for the best transformer configuration, a 4.17 percentage point improvement.

This performance gap can be attributed to the nature of the task: detecting tabular content relies heavily on structural patterns (numerical density, separator characters, column alignment) that are explicitly captured by engineered features but must be implicitly learned by transformers. In particular, standard subword tokenization tends to remove or obscure visual and layout cues—such as multiple consecutive spaces indicating column alignment and whitespace density patterns—that are critical for table detection. Furthermore, the 512-token context limit of BERT models may be insufficient to capture the full structural context of dense pages. Layout-aware models such as LayoutLM (Xu et al., 2020), which jointly encode text and spatial position, could potentially mitigate this limitation and represent a promising direction for future work. The combination of TF-IDF (capturing vocabulary) with handcrafted features (capturing structure) proves highly effective.

For production deployment, XGBoost offers additional advantages: training completes in approximately 2 minutes versus 8+ minutes for BERT fine-tuning, inference requires less than 1ms per page versus approximately 100ms for transformers, and feature importance provides interpretability for error analysis.

The two-stage pipeline effectively addresses the core challenge of processing lengthy documents with LLMs. By filtering irrelevant pages before extraction, we reduce API costs proportionally to token reduction (64–88%) and enable processing

of documents that would otherwise exceed model context limits.

5 Conclusion

We presented a two-stage pipeline for extracting structured information from Brazilian public tender documents, combining NLP-based page classification with LLM extraction. Our main contributions are:

1. A novel annotated dataset of 11,190 pages from 350 tender documents across five product domains.

2. Comprehensive evaluation of transformer and classical ML approaches for page classification, demonstrating that XGBoost with engineered features achieves 97.75% F1-score, outperforming fine-tuned BERT models by over 4 percentage points.

3. An end-to-end pipeline achieving 64–88% token reduction while maintaining 100% extraction recall against manually verified ground truth, enabling cost-effective LLM deployment for document processing at scale.

Our findings suggest that for tasks involving structural pattern recognition, classical ML with domain-specific features remains highly competitive with—and can surpass—transformer-based approaches, particularly when interpretability and computational efficiency are priorities.

To support reproducibility and future research in this domain, we will make the annotated dataset and the classification pipeline code publicly available upon acceptance.

Limitations

The proposed pipeline has several limitations. First, documents containing more than 300 items require chunked extraction due to LLM output token limitations and JSON formatting complexity. While our chunked strategy successfully handles these cases, it introduces additional processing overhead. For documents exceeding 400 items, both the baseline and pipeline approaches face increased difficulty in producing valid JSON outputs, requiring careful chunk size tuning. Second, our dataset is limited to Brazilian Portuguese procurement documents, and generalization to other languages or document types requires further validation. Third, the engineered features encode domain knowledge specific to tender documents and may require adaptation for other domains.

Acknowledgments

References

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186. Association for Computational Linguistics.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2020. BERTimbau: Pretrained BERT models for Brazilian Portuguese. In *Intelligent Systems: 9th Brazilian Conference, BRACIS 2020*, pages 403–417, Cham. Springer International Publishing.

Gemini Team, Rohan Anil, Sebastian Borgeaud, and 1 others. 2023. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. LayoutLM: Pre-training of text and layout for document image understanding.

In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200. ACM.