Structured Information Extraction from Nepali Scanned Documents using Layout Transformer and LLMs

Aayush Neupane **Aayush Lamichhane** **Ankit Paudel** **Aman Shakya**
Pulchowk Campus, Institute of Engineering, Tribhuvan University
{075bct006.aayush, 075bct004.aayush, 075bct013.ankit}@pcampus.edu.np
aman.shakya@ioe.edu.np

Abstract

Despite growing global interest in information extraction from scanned documents, there is still a significant research gap concerning Nepali documents. This study seeks to address this gap by focusing on methods for extracting information from texts with Nepali typeface or Devanagari characters. The primary focus is on the performance of the Language Independent Layout Transformer (LiLT), which was employed as a token classifier to extract information from Nepali texts. LiLT achieved F1 score of approximately 0.87. Complementing this approach, large language models (LLMs), including OpenAI's proprietary GPT-4o and the open-source Llama 3.1 8B, were also evaluated. The GPT-4o model exhibited promising performance, with an accuracy of around 55-80% accuracy for a complete match, accuracy varying among different fields. Llama 3.1 8B model achieved only 20-40% accuracy. For 90% match both GPT-4o and Llama 3.1 8B had higher accuracy by varying amounts for different fields. Llama 3.1 8B performed particularly poorly compared to the LiLT model. These results aim to provide a foundation for future work in the domain of digitization of Nepali documents.

1 Introduction

As the development on complex NLP techniques has progressed, Information Extraction from the documents has also seen lots of development. Both private as well as public companies are using different types of information extraction algorithms to streamline their business processes. Development of word representation techniques like Word2vec (Mikolov et al., 2013). and self attention mechanism (Vaswani et al., 2017) has highly impacted information extraction.These advancements allow for a more nuanced understanding of language by capturing semantic relationships and contextual information. However, there are few challenges in extracting information accurately from the documents. The challenges include first, information present in the complex document can be present in various places, there are many possible ways in which information can be organized in 2d plane, important information can be also represented in document in visual way like by underlining the words, italics word and bold word may represent different meaning, even the size of the different text can vary within single documents. Thereby making information extraction from so called Visually Rich Document (VRD) more challenging. While a good model for information extraction uses visual and layout information, along with a deeper understanding of the language, to extract information effectively, large language models like GPT (Radford, 2018) and LLaMA (Touvron et al., 2023) excel as well. Their vast scale and deep linguistic understanding allow them to perform exceptionally, even when faced with minor OCR errors, as they can often correct these mistakes seamlessly (Zhang et al., 2024).

Most of the document information extraction (IE) systems today use either sequence-tagging or sequence-generation methods. In sequence-tagging (Wang et al., 2023; Rasmus Berg Palm and Winther, 2017), each word is labeled with Inside-Outside-Begin (IOB) tags (Ramshaw and Marcus, 1995). This method helps to find and locate simple entities in the text. But, it is not easy to use these methods for extracting complex, nested entities. Pre-training of transformer based models with text and layout information (Xu et al., 2020) has been proven to be effective in a variety of visually-rich document understanding tasks due to its effective model architecture and the advantage of large-scale unlabeled scanned/digital-born documents.Models like LayoutLMv2 (Xu et al., 2022) are trained on the interaction among text, layout, and image in a single multi-modal framework. Specifically, with a two-stream multi-modal Transformer encoder.

Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025), pages 134–143
January 19, 2025.

On the other hand, sequence-generation methods (Kim et al., 2022; Powalski et al., 2021) treat extraction like generating text using autoregressive decoders (Sutskever et al., 2014). These methods can handle complex entities but cannot tell where exactly the entities are located in the document. Also, both methods need a lot of human effort to label the data correctly, which makes the process expensive.

The purpose of this study is to examine and evaluate few approaches for extracting key information from these documents. To achieve this, we experimented with LiLT (Wang et al., 2022) , which is a transformer-based model that uses layout information as well visual clues in addition to textual content. Large language models like GPT4-o (OpenAI, 2023) and Llama3.1 8b (LLaMA Team, AI @ Meta, 2024) were also evaluated, comparing the results and assessing the extraction quality.

The Language-Independent Layout Transformer (LiLT) is a model designed for structured document understanding, independent of language constraints. LiLT decouples text and layout information, optimizing them jointly during pre-training and re-coupling them during fine-tuning. This approach allows the model to learn and integrate both textual and layout features effectively. LiLT's Bi-directional Attention Complementation Mechanism (BiACM) enhances the interaction between text and layout modalities, ensuring efficient cross-modality cooperation.

We chose LiLT due to its robust pre-training on tasks like key point location, cross-modal alignment, and masked visual-language modeling. This enables it to effectively understand document layouts and content. LiLT's modular design allows for seamless integration with various pre-trained textual models, making it versatile for multilingual structured document tasks.

This work aims to improve accessibility and preservation of Nepali documents by facilitating their digitization. We hope this work lays the groundwork for future research and the development of more accurate and efficient extraction methods for Nepali texts.

2 Related Works

The field of information extraction has seen significant advancements from early rule-based systems to sophisticated machine learning and deep learning models. Initial approaches relied on rule-based methods, utilizing extensive lexicons and rules for tasks such as Named Entity Recognition (NER) and Part-Of-Speech (POS) tagging (Sarawagi, 2008; Falk Brauer and Barczynski, 2011; Deckert et al., 2011; Bertin Klein, 2019). The evolution to machine learning techniques, and subsequently deep learning models, introduced more nuanced approaches by leveraging learned token representations and reducing the need for manual feature engineering.

Information extraction from VRD is a difficult task, and there are many ways to approach it. A lot of methods break the problem into two steps. First, they use an Optical Character Recognition (OCR) service to recognize the text in the document. Then, they parse the text to find the important entities. (Xu et al., 2022) and (Appalaraju et al., 2021) handle this parsing step by using Named Entity Recognition (NER). They use a transformer encoder to label each token in the text with IOB tags, which helps to extract and locate simple entities in the document.

Other methods treat extraction as a sequence generation problem. For example, (Powalski et al., 2021) adds an auto-regressive decoder on top of a text-layout-image encoder, which is based on T5 (Raffel et al., 2023). This method helps to predict complex, hierarchical entities but does not tell us exactly where the entities are located in the document.

Self-supervised learning methods have seen a lot of development in the last several years, it is especially true in the area of natural language processing (NLP) pre-trained language models. Building on these achievements, a substantial amount of recent research has been done on structured document pre-training. For example, by adding 2D spatial coordinate embeddings to the BERT model, LayoutLM (Xu et al., 2020) improved document understanding. LayoutLMv2 (Xu et al., 2022) investigated additional pre-training tasks to better use unlabeled document data and treated visual features as unique tokens, thus improving upon the original LayoutLM model. Furthermore, LiLT introduced a more flexible and reliable way to comprehend multilingual documents that contains information in a structured format. Another notable model, LayoutXLM, extends these capabilities to multiple languages by incorporating cross-lingual embeddings to handle diverse document layouts (Xu et al., 2021). Despite these advancements, all these models have predominantly focused on English and

other major languages, with limited research addressing their performance on Nepali documents.

In addition to specialized models like LiLT, Large Language Models (LLMs) are also being used for information extraction task (Perot et al., 2024) LLMs such as GPT-4o and Llama 3.1 8B offer valuable alternatives. These models provide flexibility and ease of implementation, making them useful for initial extraction tasks. One of the great advantages of using LLMs is that they are trained on large corpus of data and they understand multiple languages making them highly suitable for document understanding task. While LLMs may not achieve the same level of accuracy as specialized models in token classification, they offer scalable solutions and can adapt to various languages with minimal retraining (Naveed et al., 2023; Brown et al., 2020).

Despite significant advancements in information extraction techniques, research focusing on Nepali documents remains sparse. Most existing studies and models, including LayoutLM (Xu et al., 2020), LayoutLMv2 (Xu et al., 2022), and LayoutXLM (Xu et al., 2021), have predominantly addressed languages with extensive resources and research focus, such as English. To address this gap, we developed a custom dataset comprising 600 scanned Nepali notices, sourced from various governmental and non-governmental institutions. This dataset includes images and manually annotated text files, providing a unique resource for evaluating information extraction techniques on Nepali documents. The creation of this dataset is crucial for assessing the performance of existing models on Nepali texts and for exploring new approaches tailored to this linguistic context.

3 Methodology

In this section, we present a systematic methodology for extracting structured data from scanned Nepali notices. We explore two distinct approaches: the first leverages the Language-Independent Layout Transformer (LiLT) model fine-tuned for token classification, and the second utilizes Large Language Models (LLMs) for information extraction. Each approach employs different techniques and tools to convert unstructured text data into a structured JSON format, which is useful for various applications such as digital archiving and data analysis.

3.1 Dataset

For evaluating the processes, we prepared a dataset of 600 notices using images and PDF downloaded from various governmental and non-governmental institutions like the Institute of Engineering, Institute of Medicine, Department of Transportation, Department of Land Management and Affairs, Ministry of Home Affairs, National Examination Board, National Disaster Risk Reduction and Management Authority, Department of Transportation Management, Ministry of Finance, and Nepal Police Personnel Record. These images were publicly available on the internet. Only the first page of the PDF was taken and converted to an image.

These documents are typically formatted in a visually structured yet information-rich manner. Most of these documents contain headers that prominently feature the issuing authority name/logo at the top, often alongside a date and authority who signed a document at the bottom. The content is predominantly textual, written in formal Nepali language, and includes specific sections such as the subject line and detailed body text. Their visual richness lies in the consistent use of logos, stamps, and proper alignment, while the textual content is dense and context-specific. These characteristics make them suitable for evaluating and comparing different information extraction techniques, especially when dealing with Nepali text, semi-structured layouts, and a mix of numeric and textual information.

We chose Subject, Date, and Signed By because they cover key information in the document. The Subject explains the purpose, the Date shows when it was issued, and Signed By identifies the authority behind it. These fields represent both structured and semi-structured data, making them useful for testing how well models extract important details.

Google OCR was used to get the text and the bounding boxes from the images. All images were manually annotated. The dataset consists of images of scanned receipts and the following text files.

- dataset_bbox.txt contains the normalized bounding boxes for the text detected by OCR.
- dataset_labels.txt contains the labels for the "signed_by", "date" and "subject" field in the IOBES format.
- dataset.txt contains the mapping between the text detected by OCR and the labels.

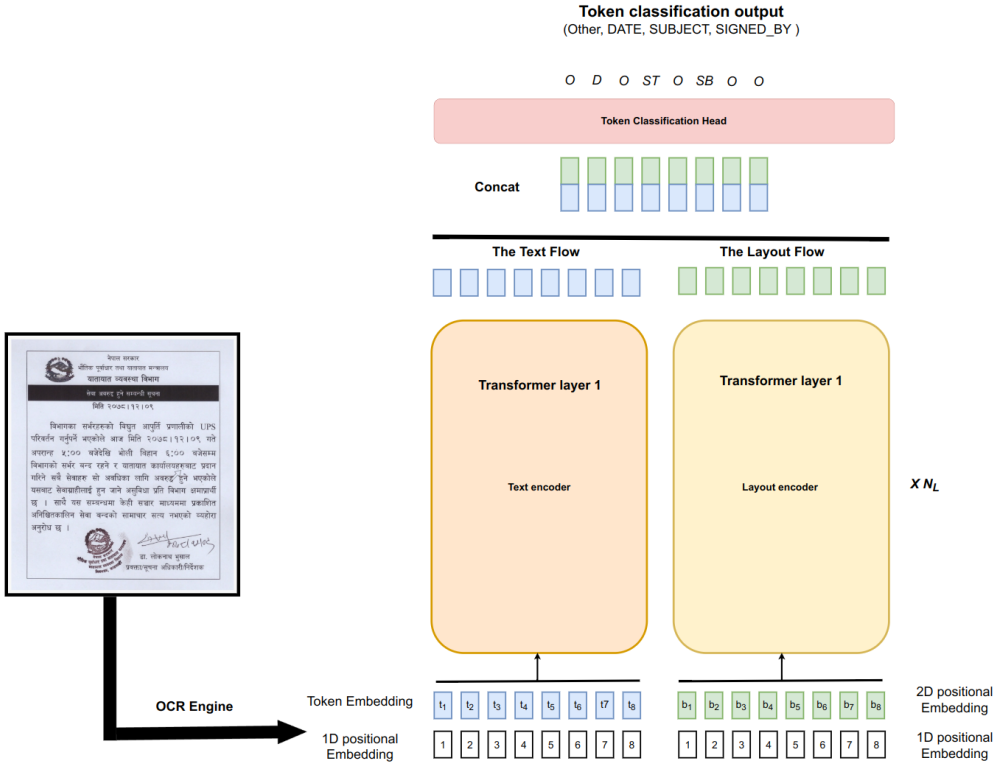

Figure 1: LiLT as a token classifier

- dataset_image.txt contains the mapping between the text detected by OCR, the bounding boxes and the original image names.

Each text file has empty newlines that separate the details of each individual image.

3.2 LiLT-based Token Classification

The LiLT-based approach is our primary methodology and focuses on leveraging the Language Independent Layout Transformer model. This model is fine-tuned specifically for the token classification task using a custom dataset of Nepali notice documents. The process involves several critical steps:

3.2.1 Model Fine-tuning

The fine-tuning of the Language Independent Layout Transformer (LiLT) model is performed with key parameters to optimize performance. Data exported after annotation were pre-processed before using it for finetuning. This pre-processing involved normalizing bounding box coordinates on a scale of 0 to 1000. The normalization of X and Y values was performed using the following equations:

$$\mathrm{X}_{norm} = \frac{X}{imagewidth} \times 1000,$$

$$\mathrm{Y}_{norm} = \frac{Y}{imageheight} \times 1000$$

For tokenization, the tokenizer from nielsr/lilt-xlm-roberta-base was utilized due to its support for multiple languages. This tokenizer specifically handles multilingual text, making it particularly suitable for scenarios that involve processing text in various languages such as low-resource Nepali language text. This tokenizer combines the Language-Independent Layout Transformer (LiLT) with XLM-RoBERTa (Conneau et al., 2019), which is a RoBERTa model trained on 100 languages. After pre-processing the data and choosing the tokenizer that supports texts in the Nepali language, model weights from the pre-trained LiLT model were loaded and the model was trained for 15 epochs with a learning rate of 5×10^{-5}. A batch size of 4 was used for both training and evaluation, considering memory constraints. The final model selected is the one with the highest F1 score on the validation set, ensuring optimal performance in token classification tasks.

3.2.2 Token Classification

After fine-tuning, the LiLT model is employed for token classification on new Nepali notice documents. In this phase, the model processes text that has been digitized through OCR. Each token in the OCR-processed text is classified into specific categories that were learned during the fine-tuning phase. This classification involves identifying tokens relevant to different information fields such as dates, subjects, and signatories. The output of this process is a set of labeled tokens, which are then used to extract and organize structured information from the raw text. We noticed that an error introduced while performing OCR is also propagated to the token classification step, as the token that has an error can sometimes be wrongly classified as a different label. This structured data is crucial for transforming unstructured documents into a format that supports efficient data organization, retrieval, and subsequent analysis. By leveraging the model's ability to classify tokens accurately, we can effectively automate the extraction of meaningful information from Nepali documents.

3.3 Large Language Model (LLM) Based Approach

The LLM-based approach represents an alternative method for information extraction, utilizing large pre-trained language models. Use of LLM falls under the sequence generator approach for information extraction. LLMs work like autoregressive decoders but modern LLMs are trained to follow instructions provided in the form of text which is also called prompt. An autoregressive decoder refers to a type of model in machine learning, particularly in sequence generation tasks, where each output token is predicted sequentially based on the previous tokens. This approach is characterized by its ease of implementation and reliance on the natural language understanding capabilities of models such as OpenAI's ChatGPT 4o and the Llama 3.1 8 billion parameter model. The methodology consists of the following stages:

3.3.1 Optical Character Recognition (OCR)

The process begins with OCR technology to extract the textual as well as layout information of scanned Nepali notices. This step converts images of text into machine-encoded text, which is necessary for further processing. Google OCR is used to ensure accurate text recognition of the Devanagari script. Google OCR works very well with documents having both English and Devnagari characters. OCR that supports both types of text in a single document is required as we have such documents in our dataset. In addition to these Google OCR also detects text that is at some angle instead of a completely horizontal text segment. Furthermore, We observed that Tesseract OCR struggled with processing documents featuring white text on a black background, while Google OCR performed effectively under these conditions. This is the reason we ended up using Google OCR for our research and it has positively impacted the performance of our extraction system.

3.3.2 Prompt Generation

Following OCR, a structured prompt is created to guide the extraction process. The prompt is designed to instruct the LLM to identify and extract key pieces of information, including the date of publication, subject matter, and signatory details. This prompt ensures that the LLM can focus on extracting relevant data accurately. Listing 1 shows the exact prompt that we used for extracting information from the document. Note that 'OCR' is a placeholder that will contain the OCR of the document that we want to extract. Moreover, it is to be noted that there is no specific reason for using this exact prompt, we tried and tested different prompts and this was a relatively good result, hence was used for our task. For instance, specifying a datatype as a comment in the SCHEMA section yielded better results.

Listing 1: Extraction Instructions

```
Please extract the following
   information from the provided
   notice: the date, subject, and
    signed_by fields. The final
   result should be in JSON
   format. The date refers to the
    publication date of the
   notice, the subject represents
    the main topic or title of
   the notice, and signed_by
   refers to the person who
   signed the notice. The keys
   should be in English, and the
   values should be in Devanagari
    script. Your output should
   strictly follow this format:

SCHEMA:
```

```
{
    "date": "",         //string
    "subject":"",       //string
    "signed_by": ""     //string
}

IMPORTANT NOTES:
If any of the fields are not
   present in the notice, you
   must leave the corresponding
   value empty.

OCR text:
{OCR}

Extraction Results:
```

3.3.3 Language Model Inference (LLM Inference)

In this stage, the LLM processes the OCR text and infers the required information based on the structured prompt. The LLM's advanced language understanding capabilities enable it to parse the text and extract relevant data points with high precision. Models like ChatGPT 4o and Llama 3.1 are employed for this inference task. To ensure consistent and accurate results, the temperature parameter for both ChatGPT 4o and Llama 3.1 was set to zero. This setting minimizes the model's randomness, leading to more deterministic and predictable outputs.

3.3.4 Decoding

The final stage involves decoding the information inferred by the LLM into a structured JSON format. JSON is chosen for its ease of use in data interchange and integration with various applications. As we can see in Figure 2, the output of LLM may contain irrelevant text in addition to the extracted information. We use a simple algorithm to detect the start and end of the JSON and extract the relevant part from this text. The structured JSON output provides a clear representation of the extracted data, facilitating its use in digital systems and databases.

4 Results

Two approaches were used for evaluating the performance of our system. For the fine-tuned LiLT model, we used classification metrics as we modeled the information extraction task as a token classification problem.

The final results, as summarized in Table 1 and Table 2, demonstrate the effectiveness of the LiLT model in accurately extracting information from Nepali scanned documents. The model achieved a precision of 89.69%, a recall of 88.20%, and an F1 score of 87.65%.

Table 1: Final Performance Metrics of the Fine-Tuned LiLT Model

Metric	Value
Precision (%)	89.69
Recall (%)	88.20
F1 Score (%)	87.65

Moreover, to effectively evaluate the information extraction system using LLMs, for each output generated by the model, a score of 0 is assigned for no match and 1 for a complete match. The Levenshtein distance is used to calculate the similarity score. Levenshtein distance is the minimum number of edits, deletions, and substitutions needed to transform one string into another.

$$similarity = 1 - \frac{L(a,b)}{max(|a|,|b|)} \quad (1)$$

where a and b are strings and L is the Levenshtein function.

The matching algorithm is especially useful for evaluating text extraction systems, as it allows for a nuanced assessment of the system's output by quantifying the degree of similarity to the expected result. We evaluated our model by taking 0.9 as a threshold and again taking a complete match. These metrics are mentioned in Table 3

These metrics reflect the overall accuracy and reliability of our information extraction system when applied to the dataset of 600 documents collected from institute notice boards.

To compare the result of LiLT model with the LLMs, we transformed the output of token classification to match the JSON output extracted using LLMs. Table 3 shows that LiLT outperforms LLMs on our dataset.

We observe that the Llama model, with its 8 billion parameters, doesn't perform as good as the GPT-4o model, which boasts a massive 200 billion parameters. This performance gap might be due to the large disparity in the size of the models,

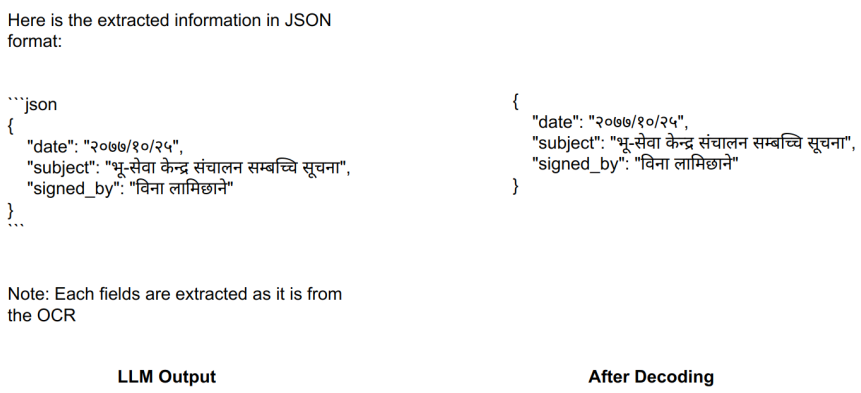

Figure 2: Filtering out JSON from LLM Output

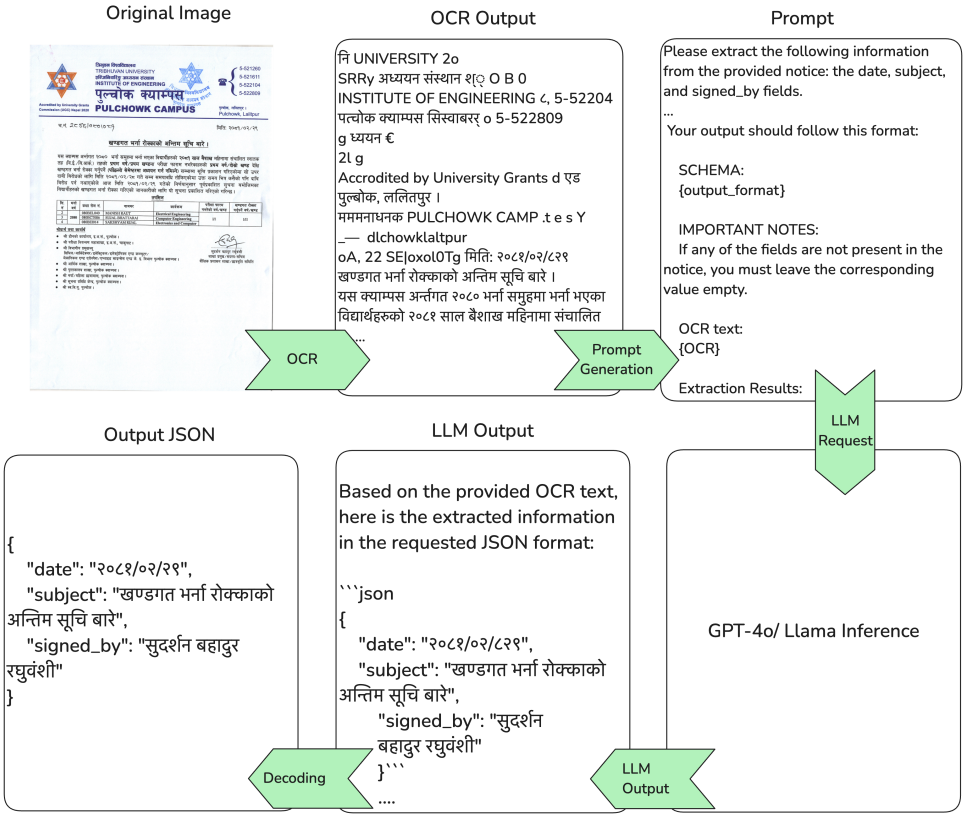

Figure 3: LLM System Process Flow Diagr In our framework, the text and layout information is first decoupled and jointly optimized during pre-training, and then re-coupled for fine-tuning

Table 2: Entity-Wise Performance Metrics of the Fine-Tuned LiLT Model

Entity	Precision (%)	Recall (%)	F1 Score (%)
Date	91.80	94.12	92.95
Signed By	90.12	89.02	89.57
Subject	87.18	84.30	85.71

the volume of training data, and the computational resources used.

Table 3: **Match Accuracy on Notices Dataset**

Label	90% Match			100% Match		
	LiLT	**GPT-4o**	**Llama-8B**	**LiLT**	**GPT-4o**	**Llama-8B**
Date	**0.81**	0.56	0.44	**0.78**	0.55	0.42
Subject	**0.93**	0.81	0.62	**0.27**	0.25	0.20
Signed By	**0.87**	0.85	0.37	**0.71**	0.66	0.30

5 Limitation

In this study, our method heavily relies on the input of text lines and bounding boxes, often generated through Optical Character Recognition (OCR) systems. This presents certain limitations, particularly in its inability to handle non-textual entities, such as images embedded within documents, which our approach does not account for. Additionally, the system is sensitive to common OCR challenges, including misinterpretations of reading order, incorrect grouping of text lines, and recognition errors. These OCR-related inaccuracies can adversely affect the performance of our model, especially in data-rich environments where precision in text extraction is critical. In addition to the Layout transformer, our approach leverages a Large Language Model (LLM) as an autoregressive decoder to extract information from the document. The LLM predicts tokens sequentially, generating outputs based on the given context. While this generative capability enables effective information extraction, it introduces a significant challenge: difficulty in localizing the extracted tokens back to their exact positions in the original document. Since LLMs are designed for token generation rather than precise token localization, mapping the output directly to specific document sections becomes complex, potentially affecting the interpretability and traceability of the extracted information.

With the LiLT transformer model, we have a limit of 512 tokens, which forces us to leave out tokens from the middle part of documents. This decision was made because, in most cases, the key information we need is found at the top or bottom of the document, while the middle part tends to be less useful for our task. By focusing on these sections, we ensure the model concentrates on what's important, though there is a chance that we might miss some relevant information in the middle. On the other hand, using LLM-based methods requires much larger models that can handle thousands of tokens to cover the whole document. While this leads to better extraction results, it also increases computational costs significantly, making these methods more difficult to scale or apply in high-volume situations. Finding a balance between token limitations and computational resources remains a challenge for our approach.

6 Conclusion and Future Works

In conclusion, our research demonstrates the effectiveness of two distinct approaches for extracting structured information from Nepali scanned documents. The fine-tuned Language Independent Layout Transformer (LiLT) model achieved high performance with a precision of 89.69%, recall of 88.20%, and an F1 score of 87.65%, indicating its robust capability in token classification and information extraction. In comparison, the large language models (LLMs) GPT-4o and Llama 3.1 8B showed variable accuracy, with GPT-4o performing generally better than Llama 3.1 8B and both GPT-4o and Llama 3.1 8B performing not as good as LiLT model. Despite their lower accuracy, LLM-based methods offer advantages such as ease of implementation and flexibility, which can be valuable for initial information extraction tasks. Integrating human feedback could further enhance the performance of LLMs and improve the overall accuracy of the system.

Thus, while the LiLT model provides higher precision, LLMs present a practical alternative with the potential for refinement and adaptation. To improve the overall accuracy of the LLM system, we can try several other methods. For instance, in the decoding phase of the LLM-based approach, we can use other approaches for example, instead of guiding the model to get the data that matches the schema provided in the prompt, we can ask the model to get all possible key-value pairs, and then pick the value corresponding to the key we are interested in.

References

Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R. Manmatha. 2021. Docformer: End-to-end transformer for document understanding. *Preprint*, arXiv:2106.11539.

Andreas R Dengel Bertin Klein. 2019. An adaptive system for document analysis and understanding. in reading and learning. pages 166–186.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Florian Deckert, Benjamin Seidler, Markus Ebbecke, and Michael Gillmann. 2011. Table content understanding in smartfix. in 2011 international conference on document analysis and recognition. *IEEE*, pages 488–492.

Adrian Mocan Falk Brauer, Robert Rieger and Wojciech M Barczynski. 2011. Enabling information extraction by inference of regular expressions from sample entities. page 1285–1294.

Geewook Kim, Teakgyu Hong, Moonbin Yim, Jeongyeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. *Preprint*, arXiv:2111.15664.

LLaMA Team, AI @ Meta. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Preprint*, arXiv:1301.3781.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

OpenAI. 2023. Gpt-4 technical report. Accessed: 2024-10-15.

Vincent Perot, Kai Kang, Florian Luisier, Guolong Su, Xiaoyu Sun, Ramya Sree Boppana, Zilong Wang, Zifeng Wang, Jiaqi Mu, Hao Zhang, Chen-Yu Lee, and Nan Hua. 2024. Lmdx: Language model-based document information extraction and localization. *Preprint*, arXiv:2309.10952.

Rafał Powalski, Łukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michał Pietruszka, and Gabriela Pałka. 2021. Going full-tilt boogie on document understanding with text-image-layout transformer. *Preprint*, arXiv:2102.09550.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Florian Laws Rasmus Berg Palm, Dirk Hovy and Ole Winther. 2017. End-to-end information extraction without token-level supervision. page 48–52.

Sunita Sarawagi. 2008. Information extraction. foundations and trends r in databases. page 261–377.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *Preprint*, arXiv:1409.3215.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jiapeng Wang, Lianwen Jin, and Kai Ding. 2022. Lilt: A simple yet effective language-independent layout transformer for structured document understanding. *Preprint*, arXiv:2202.13669.

Zifeng Wang, Zizhao Zhang, Jacob Devlin, Chen-Yu Lee, Guolong Su, Hao Zhang, Jennifer Dy, Vincent Perot, and Tomas Pfister. 2023. QueryForm: A simple zero-shot form entity query framework. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4146–4159, Toronto, Canada. Association for Computational Linguistics.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2022. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *Preprint*, arXiv:2012.14740.

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.

Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, and Furu Wei. 2021. Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding. *Preprint*, arXiv:2104.08836.

James Zhang, Wouter Haverals, Mary Naydan, and Brian W Kernighan. 2024. Post-ocr correction with openai's gpt models on challenging english prosody texts. In *Proceedings of the ACM Symposium on Document Engineering 2024*, pages 1–4.