Hebrew Donut

Project based on: **OCR-free Document Understanding Transformer** by *Geewook Kim, Teakgyu Hong, Moonbin Yim, Jeongyeon Nam,Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park* [Geewook Kim and Park, 2022]

Abstract

Visual Document Understanding (VDU) is a crucial yet challenging task in computer vision and natural language processing, requiring both text recognition and comprehensive document interpretation. Traditional approaches rely heavily on Optical Character Recognition (OCR) engines, which, despite their effectiveness, present limitations in computational efficiency, language flexibility, and error propagation. This paper explores the adaptation and extension of the OCR-free Document Understanding Transformer (Donut) for Hebrew document processing, addressing the unique challenges posed by non-Latin scripts and right-to-left languages. We present modifications to the Donut architecture to accommodate Hebrew text characteristics and develop a specialized version of the Synthetic Document Generator (SynthDoG) for Hebrew data generation. Our approach involves pretraining the adjusted model on synthetic Hebrew documents and fine-tuning it for parsing a task. This work not only demonstrates the adaptability of OCR-free models to new language domains but also contributes to the broader field of multilingual document understanding. Through a series of experiments, we evaluate the performance of our adapted model in terms of accuracy and efficiency. Our findings highlight the potential of OCR-free approaches in expanding the scope of VDU to diverse languages and document types, offering insights into the development of more flexible and efficient document understanding systems.

The code and synthetic data tool are available at: https://github.com/Matan-Lange/Donut

1 Introduction

Visual Document Understanding (VDU) has emerged as a critical task in both industry and research, with applications ranging from document classification and information extraction to visual question answering. The ability to extract and interpret information from document images such as invoices, receipts, and business cards is essential in modern working environments. However, VDU remains a challenging problem due to the complexity of tasks involved, including text recognition and holistic document comprehension. Traditionally, VDU methods have relied heavily on Optical Character Recognition (OCR) engines as a preprocessing step, focusing primarily on the understanding task using OCR outputs. While these OCR-based approaches have shown promising results, they face several limitations: High computational costs associated with using OCR engines Lack of flexibility in handling different languages or document types Propagation of OCR errors to subsequent processing stages To address these issues, we propose an adaptation and extension of the OCR-free Document Understanding Transformer (Donut) for Hebrew document processing. Our work builds upon the original Donut model, which introduced a novel OCR-free approach to VDU using a Transformer-based architecture. The key innovations of our research include: Modification of the Donut architecture to accommodate the unique characteristics of Hebrew script, including right-to-left text and distinctive character shapes. Development of a specialized version of the Synthetic Document Generator (SynthDoG) for creating Hebrew training data, enabling flexible pre-training for various languages and domains. Implementation of a two-stage training process: pre-training on synthetic Hebrew documents to learn text reading, followed by fine-tuning for specific tasks.

Our approach not only addresses the challenges of processing Hebrew documents but also contributes to the broader field of multilingual VDU. By eliminating the need for OCR, we aim to improve computational efficiency, increase language flexibility, and reduce error propagation in document understanding systems. Through extensive experiments and analyses, we evaluate the performance of our adapted Donut model on Hebrew document understanding tasks. Our findings demonstrate the potential of OCR-free approaches in expanding the scope of VDU to diverse languages and document types, offering insights into the development of more flexible and efficient document understanding systems.

2 Summary of The Original Paper

1. The paper introduces Donut (Document understanding transformer), a novel OCR-free Visual Document Understanding (VDU) model. Key aspects of the solution include:

   A Transformer-based architecture that directly maps raw input images to desired outputs without using OCR. A pre-train-and-fine-tune approach:

   Pre-training: The model learns to read text by predicting the next words, conditioned on the image and previous text context. Fine-tuning: The model learns to understand the whole document for specific downstream tasks.

   SynthDoG: A synthetic data generator for creating training data, enabling flexibility in various languages and domains. An end-to-end training process that eliminates the need for separate OCR and understanding modules.

2. The paper reports several significant findings:

   Performance: Donut achieved state-of-the-art performance on various VDU tasks, including document classification, information extraction, and visual question answering. Efficiency: The model demonstrated superior speed and memory efficiency compared to OCR-based approaches. For example, in document information extraction tasks, Donut was faster and used less memory while maintaining comparable or better accuracy. Multilingual Capability: Using SynthDoG, the authors showed Donut's ability to handle multiple languages without the need for retraining OCR engines. Robustness: Donut exhibited strong performance even on low-quality document images and showed resilience to common document image distortions, performing well on unseen document types and languages after pre-training.

3. Summary: The paper presents Donut as a simple yet effective OCR-free approach to VDU, offering advantages in performance, efficiency, and flexibility across various document understanding tasks and languages.

3 Our Project

This project explores the adaptation and extension of the OCR-free Document Understanding Transformer (Donut) for Hebrew document processing. The research focuses on modifying the Donut architecture to handle the unique characteristics of Hebrew script, including its right-to-left directionality and distinctive character shapes. A key component of the work involves adjusting the Synthetic Document Generator (SynthDoG) to create Hebrew training data and labels, enabling the model to learn from synthetically generated Hebrew document images. The project follows a two-stage approach as presented in the original paper: Pretraining the modified Donut model on synthetic Hebrew documents to enhance its ability to extract Hebrew text without relying on traditional OCR techniques. Fine-tuning the pretrained model for specific parsing tasks relevant to Hebrew document understanding. By leveraging these adaptations, the research aims to demonstrate the flexibility of the Donut model in new language domains, particularly for non-Latin scripts.

As part of our experimental strategy, we implemented a Fast (Budget) Track approach to facilitate rapid experimentation and validation using advanced personal computer resources (RTX 2070 SUPER). This track was designed to optimize efficiency in training and utilize a smaller model, allowing us to quickly iterate and refine our methods before committing to full-scale training by using DDP with 2X A100 of 160GB VRAM in total. This strategy was necessary due to the fact that the original Donut was trained on 64 A100 gpu's and we needed to be very efficient with our final training since we were very limited with time and computational resources.

For the evaluation, we used the same metrics that were used in the original paper. Pretraining Phase: Average Normalized Levenshtein Similarity (ANLS) Score - The Average Normalized Levenshtein Similarity (ANLS) score is a metric used to evaluate the performance of text recognition models. It measures the similarity between the predicted text and the ground truth text, taking into account both the correctness and the positional accuracy of characters. The ANLS score is normalized to account for varying text lengths, making it particularly useful for comparing results across different samples. A higher ANLS score indicates a closer match between the predicted and actual text, reflecting better model performance. Parsing Task: Field-Level F1 Score - The field-level F1 score is used to assess the accuracy of the model in extracting specific fields from structured documents, such as invoices. It is the harmonic mean of precision and recall, offering a balance between these two metrics. Precision measures the correctness of the extracted fields, while recall measures the completeness. The field-level F1 score is crucial in evaluating how well the model identifies and extracts key information, with a higher score indicating more reliable performance. Parsing Task: Tree Edit Distance (TED) Based Accuracy - Tree Edit Distance (TED) based accuracy is a metric used to evaluate the structural similarity between the predicted and the ground truth data in hierarchical or tree-structured information, such as the nested data in an invoice. TED measures the minimum number of edits required to transform one tree into another, considering insertions, deletions, and substitutions of nodes. The accuracy is then derived from how closely the predicted structure matches the actual one. TED-based accuracy is particularly important in assessing the model's ability to capture the hierarchical relationships and dependencies between different fields in the document.

Next, we will discuss the steps that were taken in order to achieve results within the short time we had.

4 Experiments

4.1 Experiment 1: Exploring the ViT embedding space

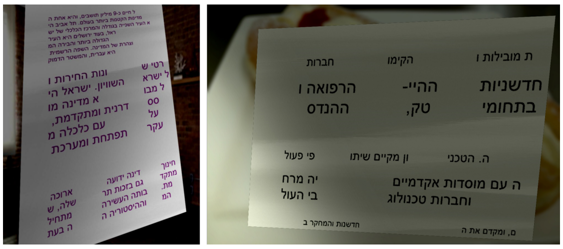
In the initial phase of our research, we conducted an exploratory analysis of the Vision Transformer (ViT) embedding space, comparing representations of Hebrew and English document images. This investigation aimed to understand how the ViT model, which forms the backbone of the Donut architecture, captures and differentiates between these two distinctly different writing systems.

We began by feeding sets of the same document images in Hebrew and English for different document types (invoices, ads, forms) through the pretrained ViT model of the Donut, extracting the resulting embedding vectors. These high-dimensional representations were then visualized using dimensionality reduction techniques such as PCA to observe potential clustering patterns and separations between the two languages. Our preliminary findings revealed several interesting insight: a tendency for the same document in both Hebrew and English to have be relatively close in the embedding space, suggesting that the ViT model captures some inherent visual differences between the scripts even without explicit language training, possibly the layouts of the documents.

This initial observation provided valuable insight for our project, hinting that the pretrained ViT model of the Donut could be used as an advanced starting point to shorten the required time of our pretrain phase.

4.2 SynthDoG-Heb - Synthetic Document Generator for Hebrew Documents

In order to effectively pretrain Donut for understanding Hebrew documents, we made modifications to the SynthDoG tool originally presented in the Donut paper. While SynthDoG was designed to generate synthetic documents for training purposes, it required several adjustments to accommodate the unique characteristics of the Hebrew language. A key modification involved altering the text direction from left-to-right (LTR) to right-to-left (RTL), which is essential for proper Hebrew text rendering. This change was not merely cosmetic but fundamental to ensuring that the generated documents accurately represented real-world Hebrew texts. Additionally, we implemented other necessary code changes to support Hebrew character sets and typography rules. These modifications to SynthDoG allowed us to generate a large corpus (based on *AlephBERT* corpus [AlephBERT, 2021]) of synthetic Hebrew documents along with their corresponding labels, overall 60k documents were generated and were divided into train (48,126), validation (5,856) and test (6,018). This tailored dataset was crucial for the pretraining phase, as it provided Donut with exposure to the specific features and structures of Hebrew text.



ל חיים כ-9 מיליון תושבים, והיא אחת ה
מדינות הקטנות ביותר בעולם. תל אביב הי
א העיר השנייה בגודלה והמרכז הכלכלי של יש
ראל, בעוד ירושלים היא העיר
הגדולה ביותר והבירה המ
וצהרת של המדינה. השפה הרשמית
היא עברית, והמשטר הדמוק

רטי ש
ל ישרא
ל מבו
סס
על
עקר

ונות החירות ו
השוויון. ישראל הי
א מדינה מו
דרנית ומתקדמת,
עם כלכלה מ
תפתחת ומערכת

חינוך
מתקד
מת.
המ

דינה ידועה
גם בזכות תר
בותה העשירה
וההיסטוריה ה

ארוכה
שלה, ש
מתחיל
ה בעת

ת מובילות ו
חדשניות
בתחומי

הקימו
ההיי-
טק,

חברות
הרפואה ו
ההנדס

ה. הטכני
ה עם מוסדות אקדמיים
וחברות טכנולוג

ון מקיים שיתו

פי פעול
יה מרח
בי העול

ם, ומקדם את ה

חדשנות והמחקר ב

Figure 1: Example of Hebrew synthetic documents.

4.3 Experiment 2: Fast (Budget) Track - Pretrain Phase Code Validation

In the Fast Track Pretrain Phase Code Validation experiment, we focused on verifying the integrity and functionality of our pretraining code and metrics using the synthetic Hebrew documents generated by our modified SynthDoG tool. This crucial step ensured that all components of our pipeline were operating as intended before proceeding to more resource-intensive stages. During this phase, we ran the pretraining process on a smaller scale, carefully monitoring various metrics to confirm that the model was indeed demonstrating learning capabilities specific to Hebrew document understanding. This validation stage allowed us to identify and address any potential issues in our code or methodology, ensuring a smooth transition to larger-scale training later in the project. The Fast Track approach proved invaluable in refining our experimental setup, providing us with the confidence to proceed with more extensive training using industry-level resources in the subsequent Full Track phase.

4.4 Experiment 3: Fast (Budget) Track - Pretrain Phase Efficiency

In our Fast (Budget) Track approach, we prioritized efficiency to enable rapid experimentation on advanced personal computer resources. The Pretrain Phase Efficiency experiment was crucial in optimizing our training process for the RTX 2700 SUPER GPU, allowing us to maximize the potential of our limited computational resources. We implemented several key optimizations to increase training speed and efficiency: Modified the Donut processor to handle smaller maximum size images, reducing memory requirements. Replaced the BART decoder with a more compact alternative, *Marian-MT* decoder [Marian-MT, 2020], and reduced the number of decoder layers, streamlining the model architecture. Employed float16 mixed precision training to optimize GPU memory usage and computational speed. Introduced mini-batches (gradient accumulation) to reduce the frequency of parameter updates, improving training efficiency. Utilized only 5 percent of the generated dataset, allowing for faster iterations while still maintaining meaningful learning.

Additionally, we explored a phased training approach by initially freezing the ViT encoder while training the decoder, then unfreezing the encoder for joint fine-tuning. This strategy demonstrated faster convergence, further enhancing our training efficiency.

This efficient Fast Track setup proved invaluable for rapid prototyping and validation of our Hebrew document understanding model before scaling up to full production training.

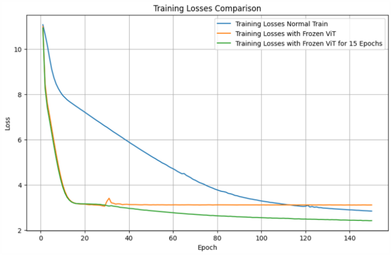


Figure 2: Training without frozen ViT vs Frozen ViT vs Frozen ViT for several epochs only.

4.5 Experiment 4: Fast (Budget) Track - Pretrain Phase Overfit

As part of our Fast (Budget) Track approach, we conducted a critical Pretrain Phase Overfit experiment. This stage was designed to test the model's capacity to learn and represent Hebrew document structures effectively, while also serving as a comprehensive validation of our synthetic data generation and code implementation. In this experiment, we deliberately allowed the model to overfit on a small subset of our synthetic Hebrew document data. This controlled overfitting served multiple purposes: It demonstrated the model's fundamental ability to learn and represent Hebrew document structures, providing early evidence of its potential for the task. The experiment acted as a first-stage validation of our synthetic data, confirming that the generated Hebrew documents contained learnable patterns and structures relevant to our ultimate parsing objectives. It served as a thorough test of our code completeness, ensuring that all components of our training pipeline were functioning correctly and interacting as intended.

By observing the model's performance as it overfit on this controlled dataset, we were able to gain valuable insights into its learning dynamics specific to Hebrew documents. This experiment provided us with confidence in our approach and highlighted areas for potential refinement before moving on to more comprehensive training phases.

The success of this overfit test was a crucial milestone in our Fast Track process, validating our core assumptions and methodologies while setting the stage for more extensive and generalized training in subsequent phases.

4.6 Experiment 5: Fast (Budget) Track - Pretrain Phase Generalization and Data Quality Validation

Following our initial overfit test, we proceeded to a crucial Generalization and Data Quality Validation phase within our Fast (Budget) Track approach. This experiment was designed to assess the model's ability to generalize beyond its training data and to scrutinize the quality of our synthetically generated Hebrew documents. For this phase, we expanded our training set to include 10 percent of our total generated data, a significant increase from the previous overfit experiment. We then initiated a training run, intentionally stopping it early to focus on the model's generalization capabilities rather than achieving peak performance. Key aspects of this experiment included: Evaluating the model's performance on unseen data from a held-out validation set, providing insights into its generalization ability. Conducting detailed error analysis to identify patterns in the model's mistakes, which could potentially reveal issues in our synthetic data generation process.

This methodical approach proved valuable, as it uncovered a significant issue in our data generation pipeline. Specifically, we discovered that numerical digits were being flipped due to the left-to-right (LTR) to right-to-left (RTL) language shift implemented for Hebrew. This finding was crucial, as it highlighted a subtle but important flaw in our synthetic document creation process.

In response to this discovery, we promptly addressed the issue by modifying our data generation code to correctly handle numerical representations in the RTL context. Subsequently, we regenerated our entire dataset with this fix implemented, ensuring the integrity of our training and validation data moving forward. This experiment underscored the importance of thorough validation in the early stages of model development, particularly when working with synthetic data for languages with different writing systems. It also demonstrated the effectiveness of our Fast Track approach in quickly identifying and rectifying potential issues before moving to more resource-intensive training phases.

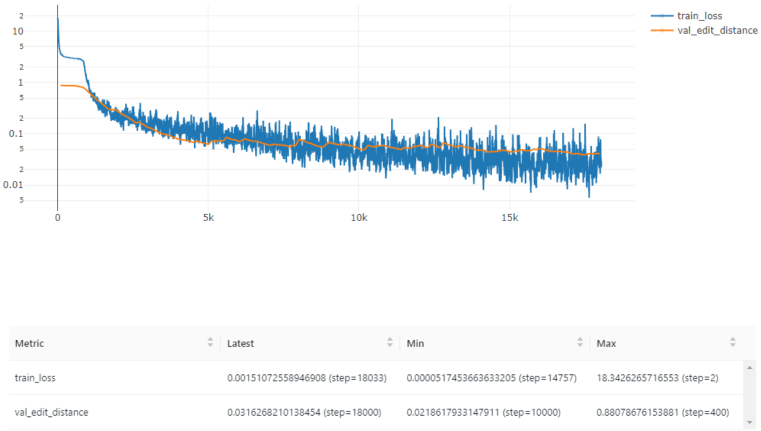
4.7 Experiment 6: Full Track - Pretrain and Evaluate

Following the successful completion of our Fast (Budget) Track experiments, we proceeded to the Full Track phase, which leveraged industry-level resources to train and evaluate our model on the entire synthetic Hebrew document dataset. This phase represented a significant scale-up in our experimental approach, both in terms of data volume and computational resources. Key aspects of the Full Track Pretrain and Evaluate phase included: Data Utilization: We employed our complete synthetic dataset for training, validation, and testing, ensuring a comprehensive representation of Hebrew document structures and variations. Computational Resources: Training was conducted using Distributed Data Parallel (DDP) on two A100 GPUs, providing a total of 160GB VRAM. This substantial increase in computational power allowed for more extensive and robust training. Model Architecture: Unlike the Fast Track, we reverted to the original Donut architecture, including the full-sized decoder with all layers intact. This decision allowed us to fully leverage the increased computational resources and potentially capture more complex document representations. Tokenizer Adaptation and embedding maps: To accommodate both Hebrew and English text, we swapped the original BART tokenizer with *Helsinki-NLP/opus-mt-en-he* tokenizer [Helsinki-NLP, 2020] and updated the SWIN tranformer's embedding map to match the expected modified embedding space shape, ensuring effective processing of multilingual content within our documents. In addition, we also used teacher-forcing, which was also used in the original paper, in order to Training Specifications:

Epochs: 3
Loss: Cross-Entropy
Total Steps: Approximately 18,000
Batch Size: 4 per device
Learning Rate: 3e-5
Gradient Clip Value: 1.0
Precision: Mixed precision with float16

Total Training Time: 11.1 hours

We implemented the efficient training pipeline developed during our Fast Track experiments, which proved scalable to this larger setup. This Full Track pretraining phase allowed us to fully exploit our synthetic dataset and train a more comprehensive model for Hebrew document understanding. The combination of our optimized training approach, increased computational resources, and the complete dataset positioned us to develop a robust model capable of handling the complexities of Hebrew document data extraction.



Metric	Latest	Min	Max
train_loss	0.00151072558946908 (step=18033)	0.0000517453663633205 (step=14757)	18.3426265716553 (step=2)
val_edit_distance	0.0316268210138454 (step=18000)	0.0218617933147911 (step=10000)	0.88078676153881 (step=400)

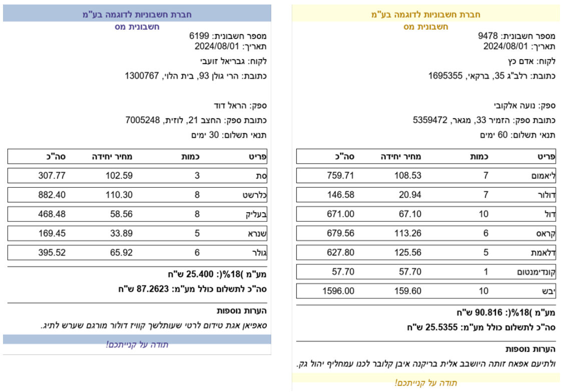
Figure 3: Learning curve log scaled.

4.8 Synthetic Parsing Data in Hebrew

After successfully pretraining Donut to understand and extract Hebrew data from documents, we proceeded to the task-specific fine-tuning phase. For our paper, we chose to focus on the parsing task, which requires the model to extract structured information from invoice documents. To facilitate this fine-tuning process, we developed a sophisticated synthetic data generation pipeline. This approach was necessary due to the scarcity of large-scale, annotated Hebrew invoice datasets and the need for a diverse, controlled dataset to effectively train and evaluate our model. Our synthetic data generation method serves several crucial purposes: Task-Specific Data: It allows us to create a large corpus of Hebrew invoices with precise annotations, tailored specifically to the parsing task. Controlled Variability: By programmatically generating invoices, we can ensure a wide variety of layouts, content, and formatting, mirroring the diversity found in real-world documents. Scalability: The system can generate thousands of unique invoices, providing ample data for both training and evaluation. Ground Truth Generation: Along with each invoice image, we automatically generate corresponding metadata, which serves as the ground truth for our parsing task. Hebrew Language Focus:

By utilizing Hebrew-specific fake data generation and right-to-left text handling, we ensure that the synthetic data accurately represents the challenges unique to processing Hebrew documents.

This synthetic data generation approach allows us to fine-tune and evaluate the Donut model on a parsing task specifically tailored to Hebrew invoices, bridging the gap between the general document understanding capabilities acquired during pretraining and the specific skills required for detailed information extraction from Hebrew financial documents.



חברת חשבוניות לדוגמה בע"מ

חשבונית מס

מספר חשבונית: 6199
תאריך: 2024/08/01

לקוח: גבריאל זועבי

כתובת: הרי גולן 93, בית הלוי, 1300767

ספק: הראל דוד

כתובת ספק: החצב 21, לוזית, 7005248

תנאי תשלום: 30 ימים

פריט	כמות	מחיר יחידה	סה"כ
סת	3	102.59	307.77
כלרשט	8	110.30	882.40
בעליק	8	58.56	468.48
שנרא	5	33.89	169.45
גולר	6	65.92	395.52

מע"מ)18%(: 25.400 ש"ח

סה"כ לתשלום כולל מע"מ: 87.2623 ש"ח

הערות נוספות

סאפיאן אגת טידום לרטי שעותלשך קוויז דולור מורגם שערש לתיג.

תודה על קנייתכם!

חברת חשבוניות לדוגמה בע"מ

חשבונית מס

מספר חשבונית: 9478
תאריך: 2024/08/01

לקוח: אדם כץ

כתובת: רלב"ג 35, ברקאי, 1695355

ספק: נועה אלקובי

כתובת ספק: הזמיר 33, מגאר, 5359472

תנאי תשלום: 60 ימים

פריט	כמות	מחיר יחידה	סה"כ
ליאמום	7	108.53	759.71
דולור	7	20.94	146.58
דול	10	67.10	671.00
קראס	6	113.26	679.56
דלאמת	5	125.56	627.80
קונדימנטום	1	57.70	57.70
יבש	10	159.60	1596.00

מע"מ)18%(: 90.816 ש"ח

סה"כ לתשלום כולל מע"מ: 25.5355 ש"ח

הערות נוספות

ולתיעם אפאח זותה היושבב אלית בריקנה איבן קלובר לכנו עמחליף יהול גק.

תודה על קנייתכם!

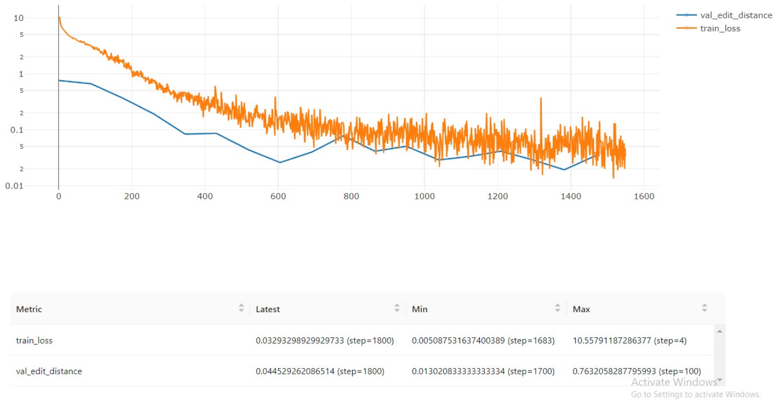
Figure 4: Example of Hebrew synthetic invoices.

4.9 Experiment 7: Full Track - Parsing Task Fine-tune

In the Full Track of our experimentation, we focused on fine-tuning our pretrained Donut model for the specific task of parsing Hebrew invoices. Due to time constraints, we opted to bypass the Fast Track testing typically used for code and data validation, proceeding directly to fine-tuning.

We employed synthetic Hebrew documents annotated with parsing labels, designed to mimic real-world invoice structures. The dataset comprised 1,000 documents, divided into 600 for training, 200 for validation, and 200 for testing. Utilizing an A100 GPU with 40G VRAM, we configured our model with a learning rate of 3e-5, a batch size of 1, and a gradient clipping value of 1.0, running for three epochs. To optimize performance and memory efficiency, we implemented mixed precision training using float16.

The fine-tuning process incorporated specific parsing tokens, enabling the model to effectively identify and extract key elements such as client information, addresses, names, dates, items, prices, quantities, and totals. This rigorous setup was essential in preparing the model for accurate and efficient parsing of Hebrew invoices in real-world applications.



Metric	Latest	Min	Max
train_loss	0.03293298929929733 (step=1800)	0.005087531637400389 (step=1683)	10.55791187286377 (step=4)
val_edit_distance	0.044529262086514 (step=1800)	0.013020833333333334 (step=1700)	0.7632058287795993 (step=100)

Figure 5: Learning curve log scaled.

5 Results

Pretraining Phase: The Average Normalized Levenshtein Similarity of 0.0296 achieved in the pretraining phase indicates a high degree of accuracy in the model's ability to understand and reproduce Hebrew text from documents. While this metric cannot be directly compared to the original paper due to its focus on task-specific fine-tuning, it suggests that the pretraining was successful in teaching the model to handle Hebrew document structures and content. Parsing Task: The F1 score of Tree Edit Distance at 0.917 for the parsing task is a significant improvement over the 0.786 reported in the original paper. This increase is noteworthy and warrants further discussion:

Synthetic vs. Real Data: - The exclusive use of synthetic data in your project, compared to real data in the original paper, may have contributed to the higher performance. Synthetic data can be more consistent and less noisy than real-world data, potentially leading to better results in controlled settings. However, this might raise questions about the model's generalization capabilities on real Hebrew invoices.

- Invoice Structure Differences: The potential differences in invoice structures between your synthetic data and the real invoices used in the original paper could explain some of the performance gap. Your generated invoices might have more consistent or simpler structures, making the parsing task relatively easier for the model.

- Corpus Limitations: The corpus used to generate synthetic text might indeed be more limited compared to the diverse real-world data and English synthetic data in the original paper. This limitation could lead to less variability in language use and potentially simpler parsing scenarios, contributing to higher performance scores.

In conclusion, while the results show promising performance, especially in the parsing task, further investigation into the model's behavior on real-world Hebrew documents and more diverse synthetic data would provide a more comprehensive understanding of its capabilities and limitations in practical applications.

6 Discussion

This project has demonstrated the feasibility and potential of adapting the Donut model for Hebrew document understanding, particularly in the domain of invoice parsing. By modifying the SynthDoG tool for Hebrew text generation, implementing a comprehensive pretraining phase, and developing a sophisticated synthetic data pipeline for fine-tuning, we have created a framework which is completely based on synthetic data that can potentially be applied to various Hebrew document processing tasks (and even additional languages). The results, particularly the impressive F1 score of 0.917 for the parsing task, suggest that our approach of using synthetic data and focusing on Hebrew-specific adaptations has merit. However, these results also raise several important points for discussion:

Synthetic Data Efficacy: While our synthetic data approach yielded excellent results, it's crucial to consider how well this translates to real-world performance. The high accuracy on synthetic data might not fully represent the model's capabilities on diverse, real Hebrew documents. Language-Specific Adaptations: The success of this project highlights the importance of language-specific adaptations in document AI. The modifications made for Hebrew, such as right-to-left text handling and Hebrew-specific synthetic data generation, were crucial to the model's performance. Pretraining Impact: The pretraining phase, resulting in an Average Normalized Levenshtein Similarity of 0.0296, appears to have set a strong foundation for the subsequent parsing task. This underscores the value of language-specific pretraining in document AI tasks. Notable Insight: During the pretraining phase, we observed an irregular second sharp drop of loss after around 2,000 steps. This is an interesting observation since its common to have an initial sharp drop of loss as the training begins which is almost always followed by plateau, according to our assumption - its when the Donut learnt to read Right-to-Left.

Open Questions and Limitations:

Real-World Performance: How well does the model perform on real Hebrew invoices? The lack of testing on a real-world dataset is a significant limitation of this study. Generalization to Other Document Types: Can the model's understanding of Hebrew documents extend to other types beyond invoices, such as contracts or financial reports? Corpus Limitations: How does the limited corpus used for synthetic data generation affect the model's language understanding and generation capabilities? Cross-Lingual Transferability: To what extent can the insights gained from this Hebrew-specific adaptation be applied to other languages, particularly those with right-to-left writing systems?

Next Steps:

Real-World Evaluation: Conduct comprehensive testing on a diverse set of real Hebrew invoices to validate the model's practical performance and identify areas for improvement. Corpus Expansion: Expand the corpus used for synthetic data generation to increase linguistic diversity and potentially improve the model's robustness. Multi-Task Fine-Tuning: Explore fine-tuning the model on multiple Hebrew document understanding tasks simultaneously to enhance its versatility. Comparative Analysis: Perform a detailed comparison with other document AI models adapted for Hebrew to benchmark performance and identify unique strengths and weaknesses. Error Analysis: Conduct an in-depth analysis of the model's errors on both synthetic and real data to guide further improvements and understand its limitations. Cross-Lingual Experiments: Investigate the model's performance on documents in languages similar to Hebrew, such as Arabic or Aramaic, to explore cross-lingual capabilities. Efficiency Optimization: Work on optimizing the model's size and inference speed for practical deployments, particularly on edge devices or in resource-constrained environments. Synthetic Data Augmentations: in order to make the Donut more robust for real documents, its possible to add augmentations such as slight rotations, shift in brightness or random noise.

In conclusion, this project has made significant strides in adapting the Donut model for Hebrew document understanding, particularly in invoice parsing. While the results are promising, they also open up numerous avenues for further research and development. The next critical phase will be to bridge the gap between synthetic data performance and real-world applicability, potentially revolutionizing Hebrew document processing in various industries.

References

[AlephBERT, 2021] AlephBERT (2021). Alephbert corpus. https://github.com/OnlpLab/AlephBERT.

[Geewook Kim and Park, 2022] Geewook Kim, Teakgyu Hong, M. Y. J. N. J. P. J. Y. W. H. S. Y. D. H. and Park, S. (2022). Ocr-free document understanding transformer. In *ECCV*.

[Helsinki-NLP, 2020] Helsinki-NLP (2020). Opus-mt english-hebrew tokenizer. https://huggingface.co/Helsinki-NLP/opus-mt-en-he.

[Marian-MT, 2020] Marian-MT (2020). Marian-mt: A fast and efficient neural machine translation model. https://huggingface.co/Helsinki-NLP.