Some Observations on the Donut Paper

Paper categories: multi-modal transformers, document AI

- The introduction section devotes a full paragraph & a diagram to explain the context of the paper by explaining how the current (i.e., prior to the paper) OCR-based document parsing pipelines work.
- Three problems of OCR-based systems are mentioned: (i) computationally expensive, (ii) not flexible across different languages and domains (document types) and (iii) error propagation.
- The paper summarizes its contributions: (i) first document AI model based on an OCR-free transformer that's been trained in an end-to-end manner, (ii) A simple pre-training scheme that enables the utilization of synthetic data -> this allows for the extension of the model to a multilingual setting. -> This is such a cool way to achieve multilinguality. **Note:** In addition, the prompts & JSON structures can be considered as contributions.
- Donut IS a language model - basically a modified BART, which is then fine-tuned (SEPARATELY) on different datasets belonging to three tasks categories - document classification, document parsing & visual question answering.
- The motivations behind each of the 3 tasks is provided: (a) *"To see whether the model can distinguish across different types of documents, we test a classification task."* (b) *"To see the model fully understands the complex layouts and contexts in documents, we test document information extraction (IE) tasks on various real document images."* (c) *"To validate the further capacity of the model, we conduct a document visual question answering task (DocVQA)."*
- This paper shows that classification can be performed with an encoder-decoder model (or for that matter with a decoder-only model) - by using special tokens. *"Unlike other models that predict the class label via a softmax on the encoded embedding, Donut generates a JSON that contains class information to maintain the uniformity of the task-solving method."*
- BART is an encoder-decoder architecture. Only the decoder part of BART is used in Donut. In other words, the encoder part of BART is simply replaced with the Swin Transformer visual encoder. Think about it. A GPT-2 style decoder cannot be used, because it has no cross-attention mechanism. So the only options were T5, BART, etc.
- The above implies that the idea of Donut is very simple: take an encoder-decoder architecture (such as T5 or BART), swap out the text encoder with an image encoder, and fine-tune. So the "pre-training" stage doesn't refer to pre-training the model from scratch. It refers to an intermediate stage of training.
- From the paper: *"The architecture of Donut is quite simple, which consists of a Transformer-based visual encoder and textual decoder modules."* **Note:** Since it's an encoder-decoder architecture with a cross-attention mechanism, there is no need to use a CLIP-like encoder (which puts image patches into the same vector space as text tokens). Any image encoder such as Swin Transformer will work. This is because the

decoder doesn't have to treat the image patches in the same way as it treats text tokens (as is the case with decoder-only approaches such as PaliGemma).

- The architecture of the visual encoder (Swin Transformer) is described in just three sentences. However, it is mentioned that the Swin Transformer architecture has been modified. See the "3.2 Setups" section in the paper. Yet, the pre-trained weights of Swin Transformer are used. From the 3.4 in the paper: "*We use all the backbones pre-trained on ImageNet.*" -> This means that the authors have managed to use the pre-trained weights (somehow) despite modifying the Swin Transformer architecture. Perhaps, they loaded the pre-trained weights one block/module at a time (instead of all at once). However, it's not mentioned whether these weights are frozen. Probably not.
- The architecture of the text decoder (BART) isn't described at all. Perhaps the authors tried out various text decoders (such as from T5 and BART), and treated each one as a black box. The only thing mentioned is that just the first 4 layers (blocks) of the decoder of BART are used, in consideration of the speed-accuracy tradeoff. **Note:** The authors have struck a balance between speed & accuracy, instead of prioritizing the latter alone.
- Public dataset for pre-training: IIT-CDIP (https://dl.acm.org/doi/10.1145/1148170.1148307) Note: Annotations aren't available for this dataset. From the paper: *"A commercial CLOVA OCR API is applied to get the pseudo text labels."*
- Public datasets for fine-tuning: *RVL-CDIP (16 different document types are covered in this dataset); CORD; Ticket; DocVQA.* **Note:** Donut is fine-tuned on each of these datasets SEPARATELY. You can find all the fine-tuned checkpoints here: https://huggingface.co/models?sort=trending&search=naver-clova-ix%2Fdonut
- For other related datasets, check out citations 26, 13, 28 & 47.
- Donut makes use of two "private industrial datasets" - "Business Card" and "Receipt". It shows some examples from them, and also reports results on them. Didn't know that this was acceptable in academic papers.
- The paper provides a nice description of each dataset, along with an example image.
- **Note:** Some image resolutions & hyperparameters are adjusted for fine-tuning & ablation studies. For example, for fine-tuning on train tickets and business cards, the image resolutions are reduced.
- The pre-training metrics aren't shared in the paper.
- The paper measures accuracy (various metrics), speed (time) & memory. **Note:** The speed is measured on a GPU that's slower than the GPU used for training.
- For classification, one metric is reported: accuracy.
- For document parsing, two metrics are reported: (i) field-level F1 score and (ii) Tree Edit Distance (TED) based accuracy. Check out citation 68. **Question:** Why not Levenshtein distance / CER? The authors have mentioned that they monitored the edit distance over token sequences (for all the datasets under all three tasks) while fine-tuning. **Note:** TED helps us verify that Donut can predict complex structures among the field information.
- **Note:** The metrics for document parsing have been reported for datasets of (i) various sizes (number of examples) and (ii) document complexities.

- For document VQA, one metric is reported: ANLS (Average Normalized Levenshtein Similarity). The score on the test set is measured via the "evaluation site" (the Document Visual Question Answering competition website presumably).
- **Note:** The metric for the DocVQA task was worse than Donut's competitors -> the authors found a subset (handwritten text) on which Donut beats its competition.
- For other related metrics, check out citations 70 & 23.
- The paper has done some error analysis (of both Donut & competitors) - and provided a few image, prediction examples that show a strength & a weakness of Donut w.r.t. the competition. The strength is handwriting recognition, and the weakness is tiny text. It's mentioned that the latter can be mitigated: "*Due to the input resolution constraint of the end-to-end pipeline, Donut missed some tiny texts in large-scale images … but this could be mitigated by scaling the input image size."*
- The authors devote a section of the paper ("Section 3.4 Further Studies") to studying the impact of several elements [(i) pre-training strategy, (ii) image backbone, (iii) image resolution and (iv) number of training examples - which is referred to as a type of *"resource"*] on Donut's downstream performance through experiments. (**Note:** In Appendix A.6, the number of GPUs is referred to as another type of *"resource".*)
- To check whether Donut is able to pay attention to proper text regions in the image, cross-attention maps in the decoder are visualized.
- To test the performance of various OCR-based systems (such as LayoutLMv2), various off-the-shelf OCRs are tested (for both accuracy & speed), and it is found (not surprisingly) that the performances of these OCR-based pipelines depend on the particular off-the-shelf OCRs being used.
- Most of the citations in the "4. Related Work" section are quite outdated. They use pipeline approaches, external OCRs, BERT for classification & extractive QA, etc. So when Donut came out, an end-to-end transformer for document AI was quite a paradigm shift.
- Research suggestions made in the paper: (i) *"Enhancing the pre-training objective could be a future work direction."* (ii) *"We believe our work can easily be extended to other domains/tasks regarding document understanding."* What are these other domains/tasks? Geometry problems & chart deconstruction are two tasks that come to mind.
- Another potential research direction mentioned in the paper: *"But, increasing the size (image resolution) for a precise result incurs bigger computational costs. Using an efficient attention mechanism may avoid the matter in architectural design, but we use the original Transformer as we aim to present a simpler architecture in this work."*