Multimodal Representations for Document Understanding

Pavel Gladkevich, David Trakhtenberg, Ted Xie, Duey Xu
{pg2255, dt2229, tx607, dx2028}@nyu.edu
Center for Data Science | New York University

ABSTRACT

Our work relies on LayoutLMv2 (Xu et al., 2021) which itself builds upon LayoutLM (Xu et al, 2020). In the first incarnation, Xu et al. pre-train visually-rich documents on text and layout features. In the second version, they are able to achieve SOTA results by incorporating image information in the pre-training stage versus vanilla LayoutLM's implementation of incorporating these image embeddings in the fine-tuning stage. The principle of their work revolves around being able to capture cross-modal interactions of visually-rich documents with the pre-training of masked visual-language models, text-image positioning, and a spatial-aware self-attention Transformer architecture. We strive to apply their multi-modal architecture to an industry setting by first imitating Xu et al.'s results then testing experiments (i.e. multi-shot document classification). We compare LayoutLMV2 to a visual baseline (ResNeXt-FPN embeddings) and a text baseline (BERT). Our experiments begin to prove the use of LayoutLMv2's architecture in industry settings.

1 Introduction

It is estimated that between 80 and 90 percent of all data in the world is stored in the form of unstructured information (Harbert, 2021). A large portion of this is in the form of scanned documents — of which modern organizations have amassed incredible amounts due to a transition towards digital business. The task of extracting meaningful information from scanned documents is complex and requires us to categorize those documents into a number of different classes of interest (e.g., memo, invoice, email, etc.). Document understanding is a crucial research area in industry. To be able to extract embedded document data affects sectors such as financial services, healthcare institutions, real estate, insurance, and government agencies. To familiarize ourselves with the range of downstream tasks across industry, we reviewed work by Intuit and Amazon. Intuit's DU platform also leverages the power of convolutional neural networks (CNNs), with models such as AlexNet, VGG-16, and ResNext101. Intuit also benefits from applying transfer learning, whereby they pre-trained their models twice, first using ImageNet and then on RVL-CDIP. They were then able to transfer these learned parameters and weights by fine-tuning on Intuit's financial documents. Similarly, we pre-train our models on RVL-CDIP and then fine-tune on ZIllow's data. For reference, our work is mentored by Zillow research scientists; while we didn't have access to their data for legal reasons, our model pipelines were still run on their datasets. They similarly make use of positional/layout embeddings as well as learn textual representations with BERT.

2 Related Work

When document analysis was in its infancy during the 90s the primary methods that dominated were rule-based approaches; however with the advances in computational speed, machine learning, and deep learning methods, the benefits of generalization to a greater variety of document classes have outweighed the advantages of simpler models. In recent years the approaches for document understanding have focused on a combination of Convolutional Neural Networks (CNN)(Hao et al., 2016), region of interest CNN methods such as Fast R-CNN (Ren et al., 2015) and Mask R-CNN (He et al., 2017), and utilizing the self-attention mechanism of Transformer architectures (Vaswani et al., 2017). For NLP and CV tasks in general there has been a shift to using pre-trained models such as BERT that can encode bidirectional representations from unlabeled text by conditioning on the left and right context of a sequence. While the BERT-like sequence labeling models can be applied in a wide variety of contexts, wherever token-level annotation is obtainable, there are limitations in the visually rich document understanding (VrDU) space. As a consequence, when tasks involving 2D documents are considered, sequential models can be outperformed by the addition of layout information either directly as positional embeddings, or through the contextualization of the spatial neighborhood (Yin et al., 2020). The model we chose to focus on implemented the former.

Vanilla LayoutLM vs LayoutLMv2

One of the first spatially aware transformer architectures in the document understanding space was LayoutLM (Xu et al, 2020), and the authors continued building improvements exemplified in LayoutLMv2 (Xu et al., 2021). Namely, LMv2 integrates image information in the pre-training stage, as opposed to vanilla LayoutLM incorporating these image embeddings solely in the fine-tuning stage. This allows LMv2 to learn cross-modal interactions between visual and textual representations. Additionally, LMv2 includes 2-D relative position representations computed in between elements of the document. This differs from LayoutLM which uses absolute 2-D positional embeddings to achieve a contextual spatial understanding. With two new training objectives (a text-image alignment strategy and a text-image matching strategy), LayoutLMv2 is able to achieve SOTA results by more capably learning cross-modal dependencies in a self-contained end to end VrDU framework.

3 Problem Definition and Algorithm

3.1 Task

Our initial goal was to understand how LayoutLMv2 achieves SOTA results in the document understanding space. We focus on document classification within the space (a standard multi-class, single label classification problem). We first established strong baselines while developing a LayoutLMv2 pipeline. In doing so, we are able to analyze at a per-class level how each model fares and hypothesize reasons why a uni-modal model may fare better than the cross-modal representations learned by LayoutLMv2 in specific instances and vice versa. In the following sections, we describe each model and their implementation, the various experiments, and consequent analysis.

3.2 Algorithms

Text Baseline

For a text-only baseline, we sought to examine classification performance when model inputs were limited to written text containing no visual or layout-oriented information. Such text is represented as a single string for each document (captured and stored in reading order; top to bottom and left to right). Our text-only model of choice was BERT, as was similarly used by the authors of the original LayoutLMv2 paper. In order to extract text data from scanned documents, we used the OCR processor Google Tesseract. From the extracted text strings the tokenization was performed with WordPiece (Wu et al., 2016), a sub-word segmentation algorithm that represents a balance between "character" and "word" delimited models. Given the verbosity of certain document examples in the dataset which contained more than 512 tokenized text pieces, examples had to be segmented using the sliding window approach (using stride of 0.8) that enabled text to be tokenized and encoded in pieces.

Visual Baseline

For the visual baseline, a scanned document image is represented as a three-layer RGB tensor. The dimensions were resized to 3x224x224 with each tensor element taking on a value between 0 (black) and 255 (white) representing each pixel. Initially, we utilized Detectron2 which employs a ResNeXt-FPN (Xie et al., 2016; Lin et al., 2017) model. Once the resized image is input into the visual encoder, a feature map is average-pooled to a fixed size. Then, it is flattened into a visual embedding sequence. A linear layer is then applied to each flattened visual embedding. Our implementation essentially isolates the ResNeXt-FPN portion of the HuggingFace implementation of LMv2. This is done by changing the output of the visual backbone with a classifier head (a linear layer) instead of feeding these visual embeddings into the Transformer architecture of LMv2.

We improved the visual baseline results by replicating LayoutLMv2's framework and feeding our visual embeddings through the transformer encoder layer which provides new visual and text embeddings. We took these visual embeddings and again added a linear layer to achieve optimal visual baseline results. This was purely done as a proof of concept that the visual embeddings from LayoutLMv2 could achieve close results, which they did.

LayoutLMv2

As mentioned, LMv2 utilizes a multi-modal spatially aware architecture. For a single page of a scanned pdf that has been processed by OCR, one of the model inputs is the tokenized text embeddings. These are formed by taking the WordPiece tokens [T] and adding 1D positional embedding (index of the token) and segment embedding (likewise the index of segment) for the final text embedding. The output of the ResNext-FPN architecture are visual token embeddings [V] that are also combined with 1D positional embedding shared with the text embedding layer and the visual segment embedding [C].

The four bounding box coordinates, width, and height are concatenated as six features that the layout embedding layer uses to construct the absolute 2D positional embeddings. The first layer input into the Transformer Layers is the fusion of these three embeddings. The transformer itself is a stack of multi-head self attention layers ending with a feed-forward neural network (FFNN) that outputs an attention score. For a single input of an encoded embedding vector, the key and query vectors are generated from the other embedding vectors by multiplying with the learned W^K and W^Q matrices. The matrix product of the resultant K, Q matrices is scaled by dividing by the dimension of the K matrix for the attention score. Then the 1D and 2D relative positions are computed between the key and query vectors and added to the attention as bias terms for the spatially aware attention score. Finally as per usual the softargmax is applied and projected on the value vectors to compute the output vectors labeled here as h. The concatenation of the output matrices from each layer as a single output matrix is sent through the FFNN for the model output on the sequence labeling task.

$$\alpha_{i,j} = \frac{1}{\sqrt{d_{head}}}(\mathbf{Q}_i)(\mathbf{W}_j)^T \qquad \alpha'_{i,j} = \alpha_{i,j} + \mathbf{b}^{(1\mathrm{D})}_{j-i} + \mathbf{b}^{(2\mathrm{D}_x)}_{x_j-x_i} + \mathbf{b}^{(2\mathrm{D}_y)}_{y_j-y_i} \qquad \mathbf{h}_i = \sum_j \mathrm{softmax}(\alpha'_{i,j})\mathbf{V}_j$$

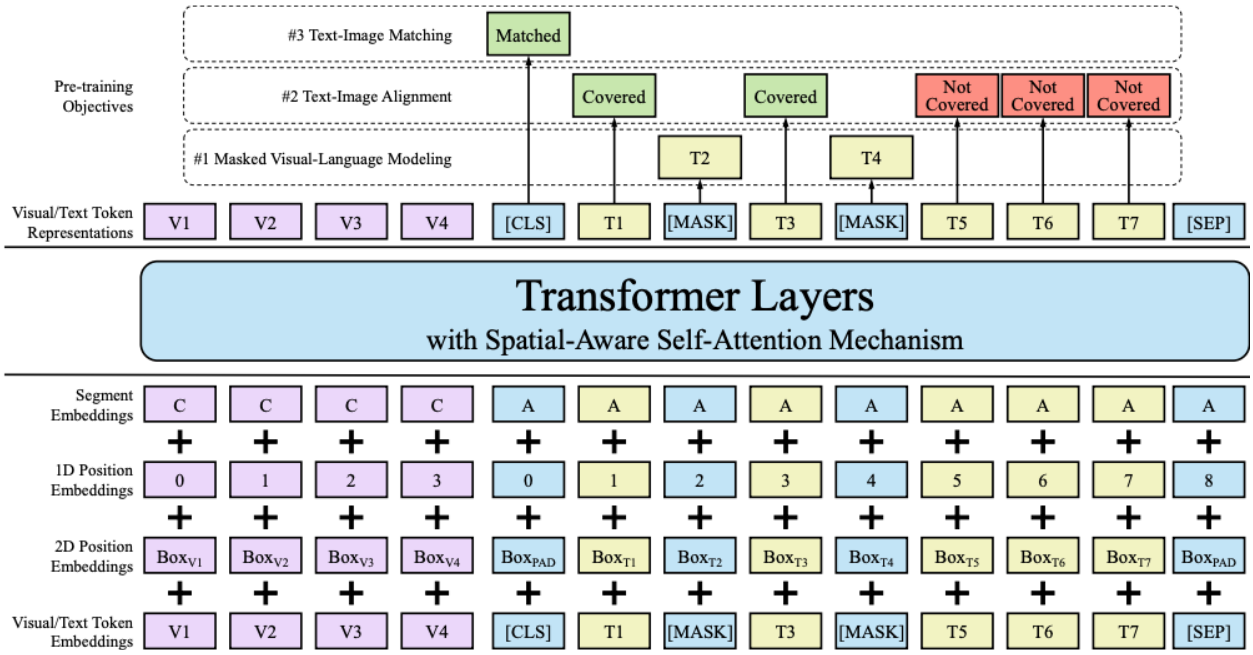

Figure 1: An illustration of the model architecture and pre-training strategies for LayoutLMv2

4 Experimental Evaluation

4.1 Data

Our experiments make use of one publicly available dataset, RVL-CDIP. Harley et al., 2015 developed a dataset that consists of 400,000 grayscale images in 16 classes. The classes are equally balanced (25,000 images per class), which lends them nicely to an evaluation metric like classification accuracy. The volume of images per train/validation/test sets are 8:1:1. The labeled documents are as follows: letter, memo, email, filefolder, form, handwritten, invoice, advertisement, budget, news article, presentation, scientific publication, questionnaire, resume, scientific report, and specification. The documents were scanned and exist as TIFF images. The Zillow dataset was 36,000 documents consisting of 4 imbalanced classes distributed unequally as 60/20/15/5. Unfortunately we could not know the identity of the class labels due to legal constraints. The OCR process was performed in an identical fashion with the assistance of our Zillow mentor.

To prepare RVL-CDIP for training (or pre-training), we made use of the off the shelf HF LayoutLMv2 Processor which combines an image feature extractor and text tokenizer following the architecture described in the previous section. Paths to the image along with the six features of image tensors, attention masks, input and token type ids, as well as bounding boxes and ground truth labels were stored in HF dataset objects. The text baseline used output from the LMv2 tokenizer alone. Pre-processing was parallelized by using 256 80GB memory cpus in sets of two that each output a single dataset file then stored as a serialized joblib file. The 128 joblib files were randomly sequentially loaded for training in and data within them was shuffled using data loaders according to each model's batch size. Unserialized the cached dataset objects combined totalled roughly 770GB.

Once the dataset's OCR was cached, we were able to leverage it for training in three different pipelines: a visual baseline, a text baseline, and LayoutLMv2 (the state of the art). These pipelines were chosen to illustrate the value of LayoutLMv2's multi-modal representation. The hypothesis tests by how much LayoutLMv2 can classify these visually-rich documents compared to the unimodal representations. Further, these experiments will allow us to analyze per document examples to better illustrate how the multi-modal model learns.

4.2 Methodology

Our three models were trained and tested on the same training / validation / test splits using the RVL-CDIP dataset. In order to account for imbalanced classes in future datasets that could be handled (e.g. Zillow's in-house scanned document dataset), a range of different classification performance metrics were selected in addition to accuracy (F1, precision, recall). Each model was trained using a single GPU and checkpointed regularly with model convergence evaluated through learning curve analysis. Checkpointed models were then evaluated on the validation and test holdout sets for final metrics. Cross entropy loss was used for all three models, although we recognize that for the Zillow dataset Focal Loss would have been a better criterion to employ due to the class imbalance. Our hypothesis was LMv2 would beat the other models in every class on CDIP.

Model	Hyperparameters
BERT	• Batch size: 32 • Learning rate: 2e-5 • Model version: BERT base uncased
ResNeXt-FPN	• AdamW Optimizer • Batch size: 128 • Learning rate: 2e-5
LayoutLMv2	• AdamW Optimizer • Batch size: 25 • Learning rate: 2e-5 * (batch size / 64)

Table 1: Hyperparameter settings for all three models

4.3 Results

4.3.1 Model Performances

LayoutLMv2 outperformed the visual and text baselines across the board. Logically on RVL-CDIP we had very similar results for F1, Precision, and recall. We also saw higher F1 scores with BERT and LMv2 as compared to the Zillow inhouse MLP and TF-IDF model which achieved an F1 of 94. The LMv2 model did beat out the baselines in every class aside from file folders, while BERT and ResNext-FPN performance varied. This points to the differing strengths of each mode. The BERT model took 10 hours to train to convergence, RESNEXT-FPN 27 hours, and LayoutLMv2 took 6-7 epochs to converge with each epoch taking around 10 hours for a little over 60 hours. Due to the multi-day convergence time for LMv2 we explored subsampling and frozen encoder layer experiments, as well as a comparison to LMv1 which we used the HF implementation for without image incorporation. LMv1 trained for around half the time and was able to achieve reasonable results, but the LMv2 training on 25% of the data was particularly impressive.

	RVL-CDIP				Zillow
Model	**Accuracy**	**F1**	**Precision**	**Recall**	**F1**
BERT	0.8559	0.8614	0.8562	0.8667	0.95
ResNeXt-FPN	0.8913	0.8662	0.8666	0.8659	0.88
LayoutLMv2	0.9321	0.9145	0.9147	0.9144	0.96

Table 2: Classification performance by model and dataset

	Model classification accuracy		
Document class	**LayoutLMv2**	**BERT**	**ResNeXt-FPN**
Letter	92.1%	89.9%	85.3%
Form	87.1%	80.9%	69.7%
Email	99.0%	97.5%	98.8%
Handwritten	95.6%	79.5%	95.3%
Ad	94.0%	66.0%	91.7%
Sci. report	89.3%	81.6%	80.3%
Sci. pub	92.5%	89.3%	89.9%
Specification	96.1%	89.6%	94.8%
File folder	95.2%	90.1%	96.8%
News article	93.6%	84.7%	92.4%
Budget	92.4%	82.8%	91.4%
Invoice	92.2%	89.0%	91.3%

Presentation	88.7%	78.9%	82.1%
Questionnaire	91.4%	85.7%	81.8%
Resume	98.5%	97.7%	95.3%
Memo	95.2%	87.2%	90.2%

Table 3: RVL-CDIP classification accuracy by document class and model

	Training data subsample			
Experiment	**Train 25%**	**Train 50%**	**Train 75%**	**Train 100%**
LMv2	90.8	91.9	92.27	93.32
Frozen	74.6	75	75.3	75.6
LMv1				88.9

Table 4: LayoutLMv2 Experiment Results

4.3.2 Error Analysis

We surveyed numerous misclassified examples to be able to detect any trends in why our models predicted certain document classes. In future work it would be beneficial to explore the decomposition of feature importance for understanding the objects a model is paying the most attention to on a given pdf image. Our subsequent error analysis is based on our own intuition and speculation.

BERT error analysis

In BERT's case, we expected the model to perform well across document classes whose language distributions were highly distinct from one another. On the other hand, we expected BERT to perform poorly where information was mostly found in the form of rich visual or layout-oriented features. For example, documents with very little text on them but lots of layout-oriented information such as forms (which are characterized by large amounts of line-delimited rectangular white space for human input); such document classes would be difficult for BERT to learn meaningful, distinguishing features.

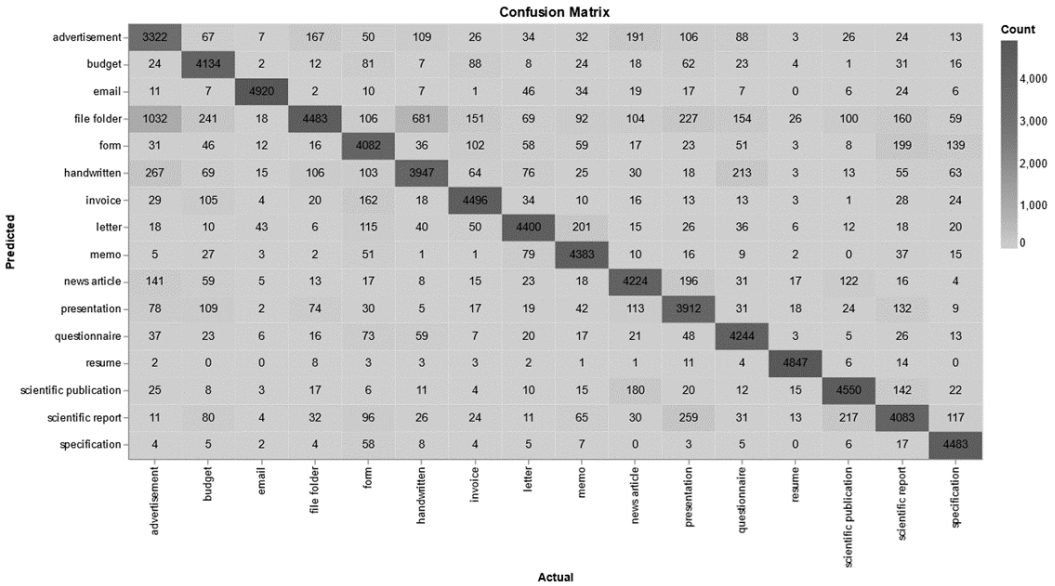

Figure 2: Confusion matrix for BERT predictions

Our confusion matrix for BERT reveals that it struggled to classify advertisements and handwritten documents the most — often confusing these with file folders. While investigating these misclassified examples we noticed that they were difficult to read; the text

extracted by the OCR for these examples tended to be extremely scarce and made little sense on their own. Noting that file folders were the most text-scarce document class across the whole dataset, it was easy to understand why BERT misclassified these as file folders. These results brought attention to an important learning: that the performance of text-based models in scenarios such as this are bottlenecked by the strength of the OCR processor used to produce text from the example documents.

ResNeXt-FPN Error Analysis

ResNext-FPN commonly struggled to classify Forms correctly, establishing a 69.7% per-class accuracy. The ResNeXt-FPN architecture misclassified Forms most commonly with Scientific Reports and Specifications. Scientific Reports share a similar visual structure in that they both have numbered lists of information that is easy to confuse, from a low resolution image standpoint. Specification sheets and Forms both are very text heavy, which the visual baseline can derive no semantic understanding from and again are easy to confuse with each other. In contrast, the visual baseline performed best on Email and File Folder, achieving a 98.8% and 96.8% per-class accuracy, respectively. These make sense as they both have highly intuitive visual structures with a common visual pattern between examples. Resumes follow a general format, with a name at the top followed by bullets of information. Also, File Folders are very similar visually as they simply contain a large amount of blank information for where the folder was scanned. Refer to figure 3 for a quantified understanding of how ResNeXt-FPN performed class by class.

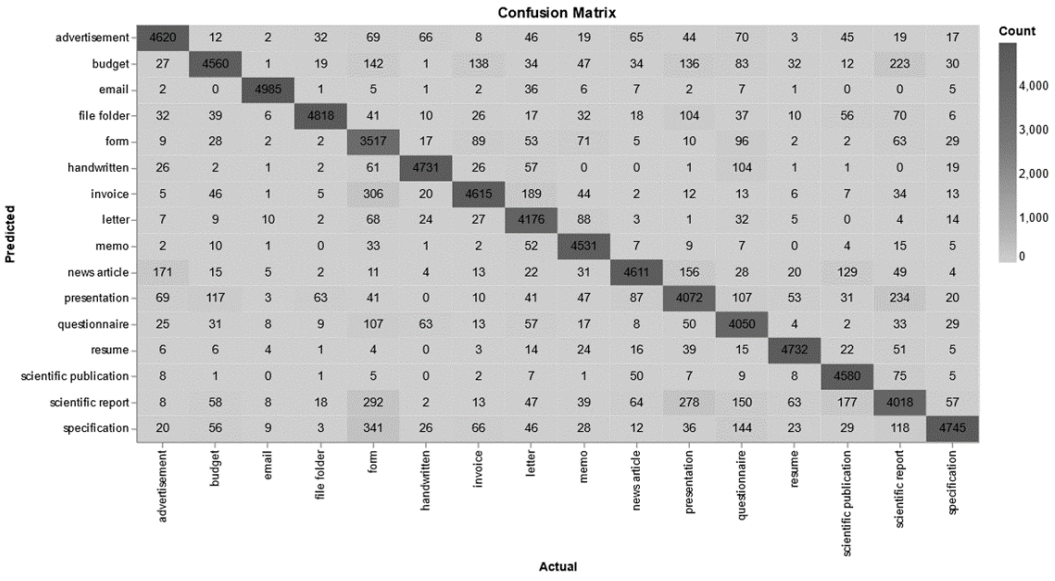

Figure 3: Confusion matrix for ResNeXt-FPN predictions

LayoutLMv2 Error Analysis

Likewise with LMv2 forms were the most problematic class at 87.1% accuracy, and were also misclassified as scientific reports and specifications, but more so as invoices. We hypothesize that one potential difference that allowed LMv2 to predict these similar classes better than the other models was the spacing between different sized tables in invoices and numerical tokens that frequently come under amount columns. Presentations at 88.7% accuracy are pretty overlapping with scientific reports (which are internal company reports), as there are likely times when the scientists were presenting their work and thus the two classes share examples. Questionnaires were misclassified as forms frequently, and while they are similar in structure, for the human eye it is easy to differentiate them since each section has a question mark at the end. LMv2 potentially overfit here on the layout of the other tokens. File Folders were the one class where LMv2 was beaten out by another model, specifically ResNext-FPN. This may have been an overfit class as there is a lack of information to learn from. We may consider dropping this class in the future, as it provides very little value.

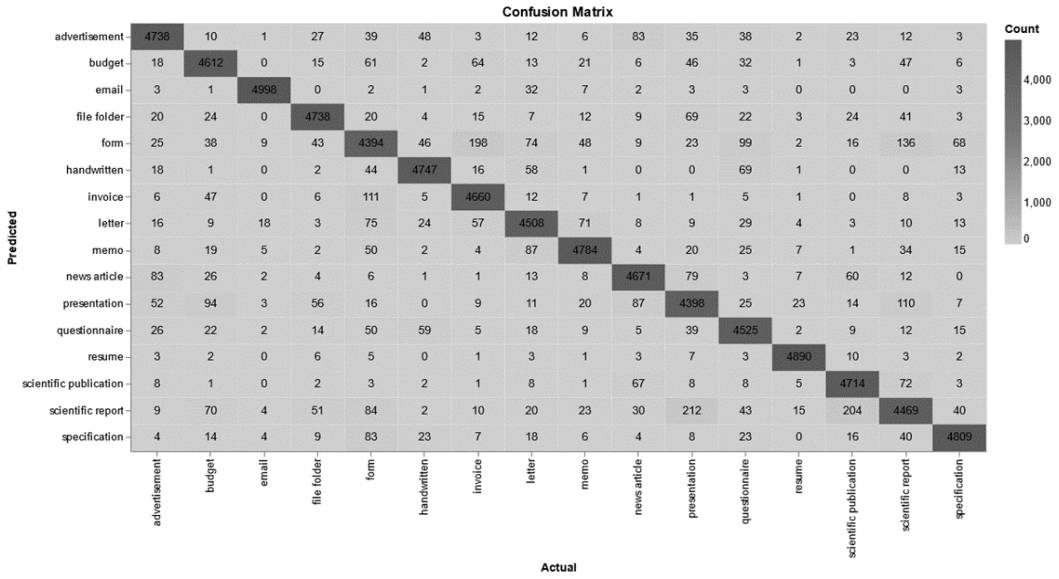

Figure 4: Confusion matrix for LayoutLMv2 predictions

4.3.3 Clustering Analysis

We wanted to visualize how our models learned to classify documents in a more intuitive manner beyond just confusions matrices. We were able to graph the learned embeddings in a 2D feature space using dimensionality reduction to inspect how our models cluster the data. We selected Uniform Manifold Approximation and Projection (UMAP) over T-distributed Stochastic Neighbourhood Embedding (t-SNE) because due to faster computational speed while giving similar visualization performance.The data for UMAP was prepared as a 10,000 example subset to run through the evaluation of all three models (BERT, ResNeXt-FPN, and LayoutLMv2). We extracted the final hidden layer outputs and averaged them to produce the embeddings, which were usually used to calculate the logits and predictions. We also tried to run UMAP for BERT using just the CLS tokens as a comparison. The embeddings are the inputs for the UMAP algorithm for dimensionality reduction.

As for how UMAP works mathematically, the algorithm uses manifold learning techniques and other ideas from topological data analysis to compute dimensionality reduction. In short, the algorithm assumes the data being fed in is used to construct a higher dimensional graph representation and optimizes another low dimensional representation to be structurally similar. UMAP uses geometrical building blocks called simplices to construct a topological representation of data. Set membership of the topological set uses the Nearest-Neighbor-Descent to quickly calculate the nearest neighbors. The algorithm then finds a low dimensional representation with the closest topological representation optimized using cross entropy metric. The optimization can also be sped up by using stochastic gradient descent.

UMAP has a handful of parameters, but the most important three that we grid searched were n-neighbors, minimum distance, and distance metric. The n-neighbors range from 1 to length of the data set and determine whether the algorithm focuses on local or global structures of the data. Minimum distance is bound from 0 to 1 and sets the minimum distance between any two points. Lastly, metric is synonymous with a distance metric. In order to generate a graphical visualization with a tight intra-group distance yet large enough inter-group for the RVL-CDIP 10k dataset, we found the best parameters for UMAP for all three models to be 50 neighbors, 0.4 distance, and a Euclidean space metric.

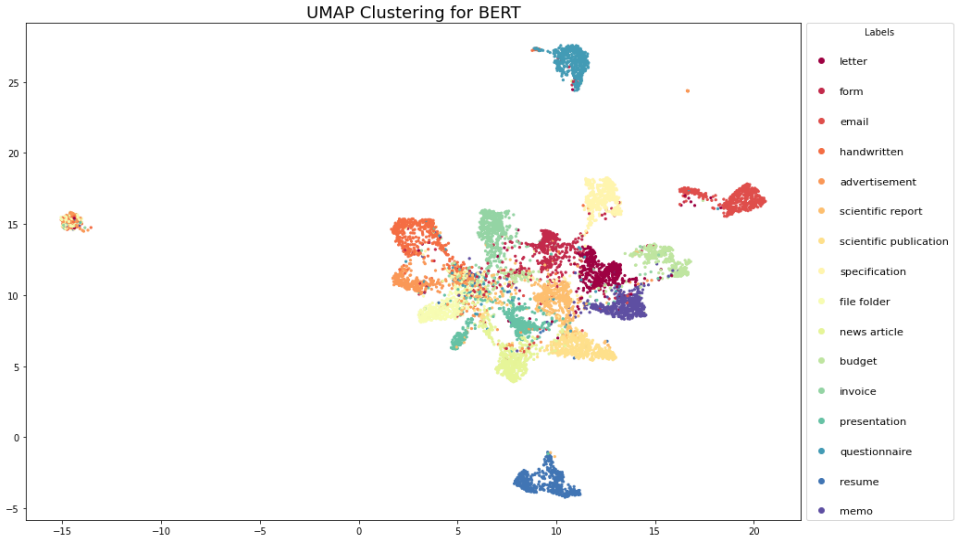

Figure 5: UMAP Visualization for Text Baseline

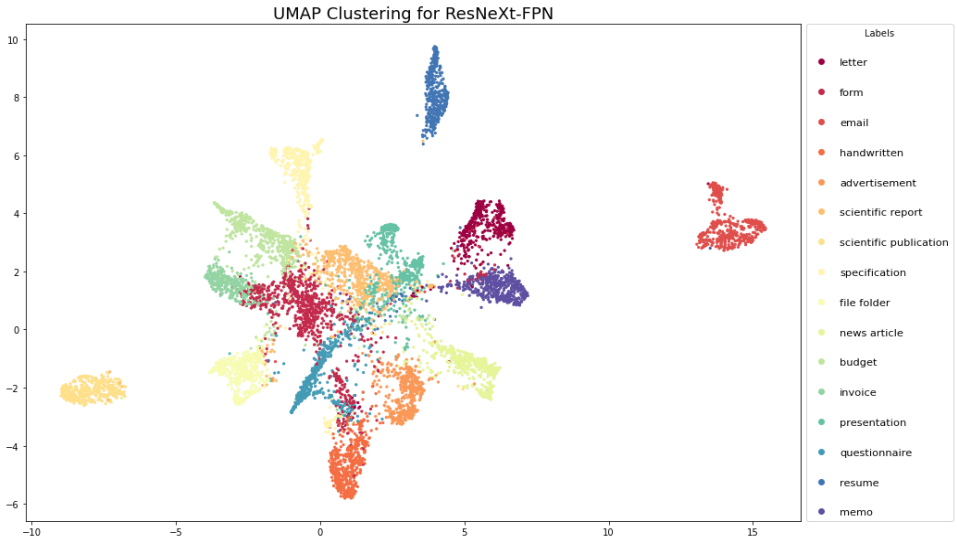

Figure 6: UMAP Visualization for Visual Baseline

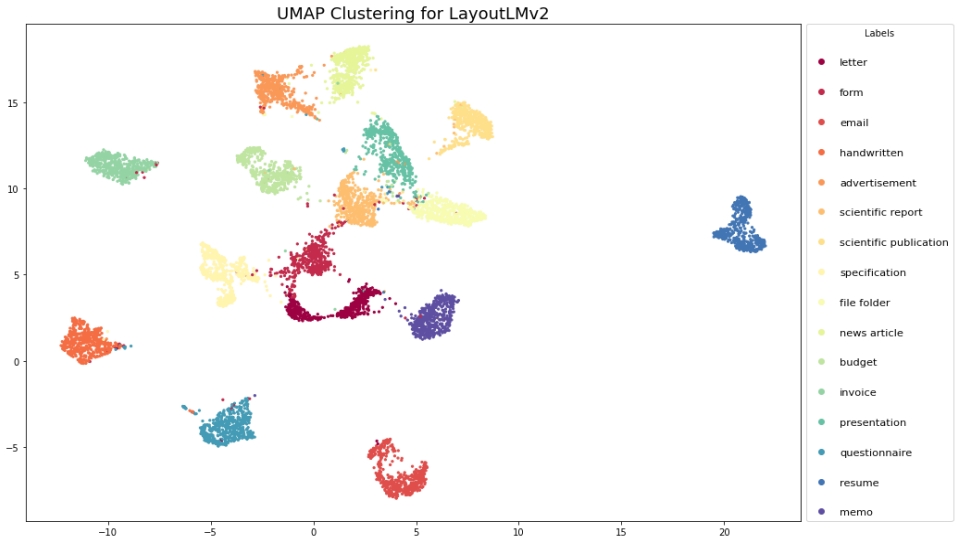

Figure 7: UMAP Visualization for Multi-modal Model

Intuitively it follows that the UMAP results were consistent with the findings of our confusion matrices. BERT did a good job of clustering resume, questionnaire, and email classes. The visual baseline was also able to accurately label emails and resumes because of the documents' distinct formatting, with the addition of scientific publications. The LayoutLMv2 model has the best UMAP visualization, mirroring the superior performance of the model where all of the clusters are distinct and at least 6 are well spaced out from the center cluster. We can see the mixture of the form class within other clusters, which is also consistent with the misclassified documents discussion from previous sections.

5 Discussion and Conclusions

We examined the document understanding space via a classification setting. Our work consisted of developing three model pipelines — BERT, ResNeXt-FPN, and LayoutLMv2. We then analyzed the model performance advantages and disadvantages of one another, which led to an understanding of various tradeoffs when one goes to productionalize these pipelines. LMv2 took over 60 hours to train, while BERT and ResNeXt-FPN reduced that time by at least 50%. The tradeoff to consider here is that of time and computational power. Another consideration in this tradeoff realm is that ResNeXt-FPN does not need an OCR processor, and a major takeaway we had is that model performance is bottle-necked by the quality of the OCR processing. Some of the methods that we could use to improve this process is applying better denoising, increasing the image contrasts, and utilizing de-skew techniques such as rotating and flipping the image, then taking the most likely OCR by values by perplexity. We believe that this could be one of the reasons our accuracies are below those reported in the LMv2 paper.

More generally, we were not surprised to learn that in most settings LMv2 beat out BERT and ResNeXt-FPN. LMv2 incorporates BERT and ResNeXt-FPN's architectures in its multi-modal model, so intuitively it should outperform them. At its crux, LMv2 is able to learn cross-modal dependencies from visually-rich documents, which proves to be invaluable since layout and text/visual cues can be widely changing from document to document. LMv2 is also very apt to transfer its learned representations from one dataset (i.e. RVL-CDIP) to an entirely different one (i.e. Zillow's in-house data). The use case of more complex models such as LMv2 stems from its superior performance on classes with lots of detail and varied structure. These classes frequently overlap and thus differentiating between them can be a challenge for humans and simpler models alike.

One of the potential future applications of our work that we would like to explore more given time is one or few shot learning. This is the task of constructing a model that can quickly learn with few labeled examples. Using the same process as for UMAP, our mentor extracted our checkpointed model's hidden state representations and used it as input into a kNN classifier as features. With only 50 labeled samples for unseen classes (classes not represented in training data) this achieved a F1 score of 61% when evaluated on the Zillow dataset. It is possible this could be improved by pre-training on domain data, using representations from multiple layers by taking the maximum, averaging, concatenating, or the [CLS] token representation alone (Devlin et al., 2019). More exploration needs to be done into these other variations. This could be done by re-using RVL-CDIP with a subset of classes removed during the fine-tuning phase. Other options for improving one or few shot models would be exploring meta-learning tasks such as siamese learning, so training multiple identical networks that share the same weights. Then the hidden output of these networks is converted from a classification task to a similarity task. Loss functions used for this such as contrastive, triplet, or circle loss minimize intra class similarity distance, or distance between samples within the same group. The CDIP dataset could be re-used for this task if models were fine-tuned with the omission of some classes.

Another problem would be the splitting of pdfs that consist of multiple document types for variable page run lengths into subdocument groups, or document splitting. This could be approached through either the binary classification task of start pages, or the multi-label task of start pages in addition to subdocument group labels. It would be necessary to utilize datasets that have page ordering information or a dataset with full pdfs. Publaynet or Scanbank are such examples. Again, the hidden state embedding output of our models could be used here as features into a secondary classifier such as K-means.

For Zillow, and many other firms that rely on processing large quantities of digital documents, the task of labeling examples can be an expensive and manual process. Using an active learning paradigm, a learner can intelligently query the most informative instances. In this way, the learner will self-label unlabeled documents, with a level of uncertainty. If a company can label a small amount of their training set, an active learner can dynamically build a labeled training set on its own. This has huge implications for future work. One framework that can be implemented is modAL (built on top of scikit-learn) which allows for "modularity, flexibility and extensibility" (Danka 2018).

6 Lessons learned

We found this experience to be both very rewarding and challenging. For the first few weeks, our mentor had us read relevant literature in the document understanding space. Having little experience using neural network architecture, the readings were difficult to grasp at first but with time, we were able to learn a great deal. This taught us the importance of staying up to date with the state of

the art, whether one is in industry or academia—LayoutLMv2 was only published May 2021! We learned the ins and outs of training complex architectures on large amounts of data. This includes things like managing our time and resources effectively as a job could take 3+ days on the cluster to finish; the significance of running several experiments (i.e. ablation studies); and the importance of version control when programming in a team/group setting. Aside from the domain experience (document understanding) we've gained as well as the relevant deep learning experience (i.e. CNN, ResNeXt-FPN, etc.), we have learned the importance of presenting our data science learnings/findings to an audience and distilling complex information to be more consumable.

7 References

Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In International Conference on Document Analysis and Recognition (ICDAR), 2015. https://arxiv.org/abs/1502.07058

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, May 24). Bert: Pre-training of deep bidirectional Transformers for language understanding. arXiv.org. Retrieved December 2021, from https://arxiv.org/abs/1810.04805

Hao, L., Gao, L., Yi, X., & Tang, Z. (2016). A Table Detection Method for PDF Documents Based on Convolutional Neural Networks. 2016 12th IAPR Workshop on Document Analysis Systems (DAS), 287-292. https://www.semanticscholar.org/paper/A-Table-Detection-Method-for-PDF-Documents-Based-on-Hao-Gao/06bf934004b6f93711298f905b1e447683a8d0b9

Harbert, T. (2021, February 1) *Tapping the power of unstructured data.* Ideas Made to Matter. https://mitsloan.mit.edu/ideas-made-to-matter/tapping-power-unstructured-data

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. 2017. Mask R-CNN. CoRR abs/1703.06870 (2017). arXiv:1703.06870 http://arxiv.org/abs/1703

Leipeng Hao, Liangcai Gao, Xiaohan Yi, and Zhi Tang. (2016). A Table Detection Method for PDF Documents Based on Convolutional Neural Networks. 2016 12th IAPR Workshop on Document Analysis Systems (DAS) (2016), 287–292 https://www.semanticscholar.org/paper/A-Table-Detection-Method-for-PDF-Documents-Based-on-Hao-Gao/06bf934004b6f93711298f905b1e447683a8d0b9

McInnes, L., Healy, J., & Melville, J. (2020, September 18). *UMAP: Uniform manifold approximation and projection for dimension reduction*. arXiv.org. Retrieved December 2021, from https://arxiv.org/abs/1802.03426

Ouali Y., Hudelot C., Tami M. (2021) Spatial Contrastive Learning for Few-Shot Classification. In: Oliver N., Pérez-Cruz F., Kramer S., Read J., Lozano J.A. (eds) Machine Learning and Knowledge Discovery in Databases. Research Track. ECML PKDD 2021. Lecture Notes in Computer Science, vol 12975. Springer, Cham. https://doi.org/10.1007/978-3-030-86486-6_41

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (2015), 1137–1149 https://arxiv.org/abs/1506.01497

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017) https://arxiv.org/abs/1706.03762

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R. (2019). Detectron2. GitHub. Retrieved December 2021, from https://github.com/facebookresearch/detectron2

Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., Zhang, M., & Zhou, L. (n.d.). Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. – arXiv Vanity. Retrieved December 8, 2021, from https://www.arxiv-vanity.com/papers/2012.14740/.

Yin, P., Neubig, G., Yih, W.t., Riedel, S.: TaBERT: Pretraining for joint understanding of textual and tabular data. In: ACL (2020) https://arxiv.org/abs/2005.08314

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation https://arxiv.org/pdf/1609.08144v2.pdf

8 Student contributions

Pavel: Loaded data on cluster, OCR, LayoutLM models, Misclassified examples, Poster, Paper

David: ResNeXt-FPN model implementation and experimentation, Poster, Paper

Ted: Data splits, Initial ResNeXt-FPN model implementation, Poster, Paper, UMAP

Duey: BERT model implementation and experimentation, Poster design, Paper