Few-Shot Entity Recognition from Document Images

Surya Krishnamurthy, Sai Chandra Pandraju
Northeastern University, Boston

Abstract

Entity recognition on text in visual documents is an important problem. While neural networks perform well on the task with large, annotated data, annotating data can be challenging. In this project, we perform few-shot entity recognition on visual documents, performing entity recognition with a small sample of data for unseen entities. We present a prototypical network using pretrained transformers as backend and its accuracy results on the CORD dataset. Our best performing ProtoNet-LayoutLM model achieves an accuracy of 81% on 5-way 5-shot entity recognition. We report the model performance on various few shot settings.

Introduction

Entity recognition is an important and fundamental task in Visual Document Understanding (VDU), an umbrella term for various tasks associated with processing information from unstructured data in document images such as invoices, forms, and contracts (Cui, Xu, Lv, & Wei, 2021). The goal is to automatically process large amounts of visual documents and extract structured information that can be used for various applications, such as document classification, data extraction, and data entry. In entity recognition, we identify text span roles in the document, such as a table, figure, date, receipt ID etc. It is a multi-modal problem with intersections in computer vision and natural language processing.

Recent successes of transformers have led to the development of various neural network architectures that perform well on VDU tasks with sufficient data. However, their performance under the limited data regime is an active area of research. Particularly, approaches for VDU tasks like entity recognition in low-data regime are rare. Due to the diverse nature of documents, finding training datasets like those in various use cases is challenging. Manually labelling these images can also be extremely cumbersome, error-prone, and expensive. Therefore, approaches to developing models with limited annotated data can be very valuable. We will develop models that perform well in the few-shot setting and explore the performance of neural network models. In **few-shot learning**, we will have a small support set of documents of size K containing some B-labelled entities (hence B-way K-shot), which we will use to perform entity recognition on a query set containing unlabeled documents. We use prototypical networks (Snell, Swersky, & Zemel, 2017), a flexible and popular metric learning algorithm for few-shot learning.

However, developing few-shot learning methods for visual documents can still be challenging as it is very difficult to find datasets that are large and contain several classes. As a result, models are prone to overfitting and fail to generalize. Modern advances in unsupervised pretraining of large neural networks allow us to extract informative features from images without explicitly training on large, annotated data. These feature extractors are excellent priors for models in various downstream tasks. We will leverage these pretrained feature extractors with prototypical networks to build efficient few-shot learners. Our code is publicly available at https://github.com/SuryaThiru/ProtoIE.

Dataset

CORD

A **CO**nsolidated **R**eceipt **D**ataset (Park, et al., 2019) containing 11000 images. 5 superclass and 30 subclass labels (recently, some classes were redacted due to privacy issues) are present. We will be working with 23 of these subclasses for our study, which excludes classes that occur sparsely in the dataset (Figure 1 Label distribution). The annotated dataset contains OCR extracted bounding boxes and corresponding texts which are later annotated with the appropriate labels. We use a dataset version from huggingface datasets for our work (katanamlcord).

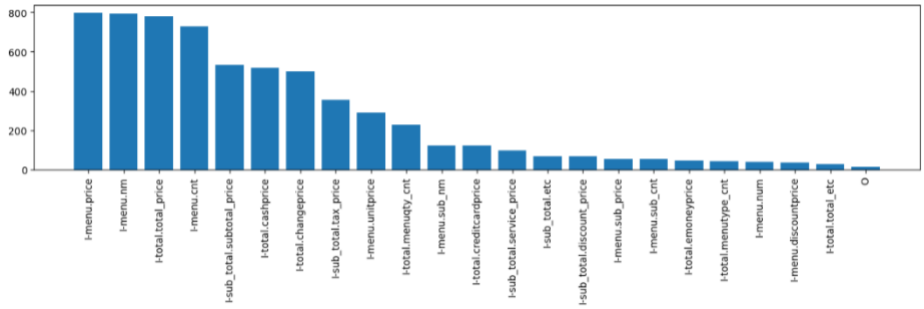

Figure 1 Label distribution

Figure 2 A sample receipt from CORD

Methods

Prototypical Networks

We use the framework described in (Snell, Swersky, & Zemel, 2017) to develop models for few-shot learning. Their simple inductive bias allows us to flexibly integrate modern pre-trained transformers in the few-shot learning regime. The approach computes the centroid of points in a neural network's embedding space as a class's single prototype representation. Classification is then performed by simply finding the nearest cluster as show in Figure 2.

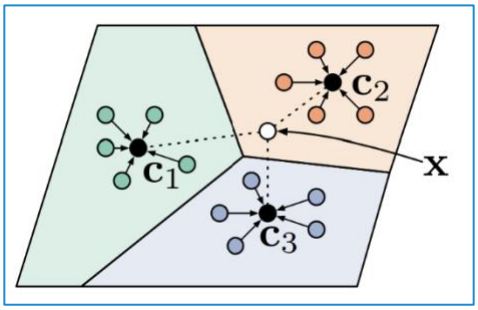

Figure 2 Prototypical networks for few-shot classification (image from original paper).

Given a support set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ with $x \in R^D, y \in R^1$, and an embedding function $f_\phi: R^D \rightarrow R^M$ with parameters ϕ, the prototypes are computed as:

$$c_k = \frac{1}{|S_k|} \sum_{(x_i, y_i) \in S_k} f_\phi(x_i)$$

The class distribution is then predicted using a distance function $d: R^M \times R^M \rightarrow [0, \infty)$ (in our case, the Euclidian distance) with a SoftMax function:

$$p_\phi(y = k|x) = \frac{\exp\left(-d\left(f_\phi(x), c_k\right)\right)}{\sum_{k'} \exp\left(-d\left(f_\phi(x), c_{k'}\right)\right)}$$

Once the SoftMax output is computed, the model is trained using cross-entropy loss:

$$J(\phi) = -\sum_k y_i \log p_\phi(y_i = k|x_i)$$

We use large pre-trained models as backends for our prototypical networks. Particularly, we use LayoutLMv3 (Huang, Lv, Cui, Lu, & Wei., 2022) and LiLT (Wang, Jin, & Ding, 2022), which are both transformers that are pre-trained on large corpus of visual documents. They act as excellent feature extractors owing to the representations learnt with the pretraining objective.

Backends:

LayoutLMv3

LayoutLMv3 is a transformer-based deep learning model designed for document image understanding. It is a multimodal model that has been pre-trained using a word-patch alignment objective, enabling it to learn cross-modal alignment by predicting whether the corresponding image patch of a text word is masked. It is pretrained

using a large scale scanned document image dataset containing about 11 million document images with 3 objectives - Masked Language Modeling (MLM), Masked Image Modeling (MIM), and Word-Patch Alignment (WPA). LayoutLMv3 is a versatile pre-trained model suitable for text-centric and image-centric Document AI tasks thanks to its simple unified architecture and training objectives.

LiLT

Language **i**ndependent **L**ayout **T**ransformer (LiLT) addresses the language dependency issue inherent in most current document understanding models by proposing a straightforward yet effective approach. This approach involves pretraining on structured documents of a single language and then fine-tuning directly on other languages using corresponding off-the-shelf monolingual/multilingual pre-trained textual models. During pretraining, text and layout information are initially decoupled and jointly optimized before being recoupled for fine-tuning. In addition, LiLT proposes a novel bi-directional attention complementation mechanism (BiACM) to ensure that the two modalities have sufficient language-independent interaction and to enhance cross-modality cooperation. Alongside the Masked Visual Language Modeling (MVLM) technique, the model proposes key point location (KPL) and cross-modal alignment identification (CAI) tasks as pretraining objectives.

Baseline Nearest Neighbors

We use a 5 nearest neighbor classifier as a baseline model. The approach simply compares the points in query set with points in support set and takes the majority class of the 5 nearest points. We obtain feature embeddings from a pretrained LayoutLMv3 model and use it for computing distances.

Training and Evaluation

We use the standard episodic training approach to train and evaluate our few-shot learners, which differs from the traditional supervised training approach. In each episode, a support set, and query set is sampled from the original data containing a randomly chosen subset of all our labels. Since we treat the entity recognition task as a token classification problem, model is trained by minimizing an appropriate loss that captures how well the model predicts the classes of the query set. Several episodes are run across multiple epochs. During the testing stage, the same procedure is used to evaluate the performance without any updates to the model. To fairly evaluate the performance of our models, we use mutually exclusive sets of training (13 classes) and testing labels (10 classes).

The steps involved in each episode is enumerated below:

1. Select a random subset of classes.
2. Select support examples for the selected classes.
3. Select query examples for the selected classes.
4. Compute loss/accuracy from the predictions on the query set and its ground truth values.
5. Update model using the loss obtained.

Note that our prototypical networks use cross-entropy loss as described in the methods section. Following (Snell, Swersky, & Zemel, 2017) we use the same sample size for support and query sets in training and testing episodes.

We evaluate the models using the accuracy (or F1 micro) metric.

We implement our algorithms using PyTorch (Paszke, et al., 2019), extending the pretrained models from huggingface. The models were trained on an Nvidia P100 GPU.

Results

Model Performance

We trained our ProtoNet model with various pre-trained backends and a feed-forward neural network head. Due to limitations in computational resources, we only train the parameters of the feed-forward head, keeping our backend parameters frozen. From experimentation, we determined the configuration shown in Table 1 to work well. We use the same configuration for all the analysis presented in this section.

Hyperparameter	**Value**
Epochs	10
Episodes	100
Optimizer	Adam
Learning rate	0.005
Learning rate schedule	Gamma 0.5, step size 2

Table 1 Hyperparameters configuration

The comparison of test accuracy between our methods on the 5-shot 5-way settings is shown in Table 2. We use a feed-forward head with 2 hidden-layers with 256 and 64 nodes respectively. The LayoutLMv3 model, with better use of multi-modal signals from the input image and OCR texts, outperforms the other models in classification accuracy. The superior performance of the nearest neighbor baseline over the random baseline shows the ability of the pretrained layoutlmv3 model to capture essential features of visual documents in its representations without any fine-tuning on unseen dataset.

Method	**Accuracy**
ProtoNet – LayoutLM	**81**
ProtoNet – LiLT	80
Baseline – nearest neighbors	52
Baseline – random	20

Table 2 Comparison of methods

Since we only train our feed-forward head, we record the performance of different configurations of the network. We use the layoutlmv3 backend for the experiments shown in Table 3 and evaluate under the 5-way 5-shot recognition setting. We observed that using wider networks results in an overfit model that performs poorly on the test set.

Neural network configuration	**Accuracy**
512-256	78
256-64	81
128-32	80
256-128-64-32	80

Table 3 Performance of ProtoNet-LayoutLM with different feed-forward heads

Few-shot settings

To assess the ability of our approach to work with different number of classes, we analyzed the performance of our best performing model with different number of classes. Due to the limitations in the available number of classes, we test up to 7 classes under the 5-shot setting. The results are shown in Table 4. The models perform well even with a higher number of classes.

B-way	Accuracy
3	83
5	81
7	79

Table 4 Performance of ProtoNet-LayoutLM with different number of classes

Similarly, the performance across different sizes of the sample set is analyzed. Following the original protonet paper, we keep the size of the support set and query set the same. Results are shown in Table 5. As expected, the performance improves with the increase in the size of the samples per class. However, very poor performance was observed for single shot recognition.

K-shot	Accuracy
1	20
5	81
7	84

Table 5 Performance of ProtoNet-LayoutLM with different sample size

Out of distribution performance

Apart from the challenges associated with training models under limited supervision, several challenges present themselves when working with documents in the wild. Documents could be physically captured through an input device like a camera or scanner, or present in files as pdfs or other image formats. Documents also contain aberrations that can be from errors in printing, scanning, or capturing the document. Therefore, receipts vary significantly in quality, formatting, and noise. To assess our model's ability to perform entity recognition on out of distribution data, we used our model trained on the CORD training set and evaluated it on episodes on the SROIE dataset (Huang, et al., 2019), which contains 4 labels (company name, address, total, date) which are quite different from the CORD dataset. However, it is also a receipt dataset with minor variations in formatting, language and quality. Using the ProtoNet-LayoutLM model, we achieve a **66%** accuracy on entity recognition in SROIE dataset. We hypothesize that the drop in performance is due to the feed-forward network acting as a bottleneck in our approach, overfitting on the CORD dataset. Re-training the network end-to-end, including the layoutlmv3 backend could improve the performance of the model. Incorporation of more generalization tricks could also improve the performance of our model.

Representation learning

An essential attribute of a deep learning-based metric learning algorithm is their ability to capture efficient representations of the input features. Particularly, since the task requires the model to capture both the language and image features, good representations will help the model capture inter-entity and intra-entity relationships. To assess if our model improves the feature representations of our original pretrained backend, we decompose our hidden representations to 2 dimensions using TSNE and qualitatively evaluate the latent encoding of the tokens present in a random sample of 15 documents. Figure 43 shows an improved density in the latent space for similar entities and better separation across different entities.

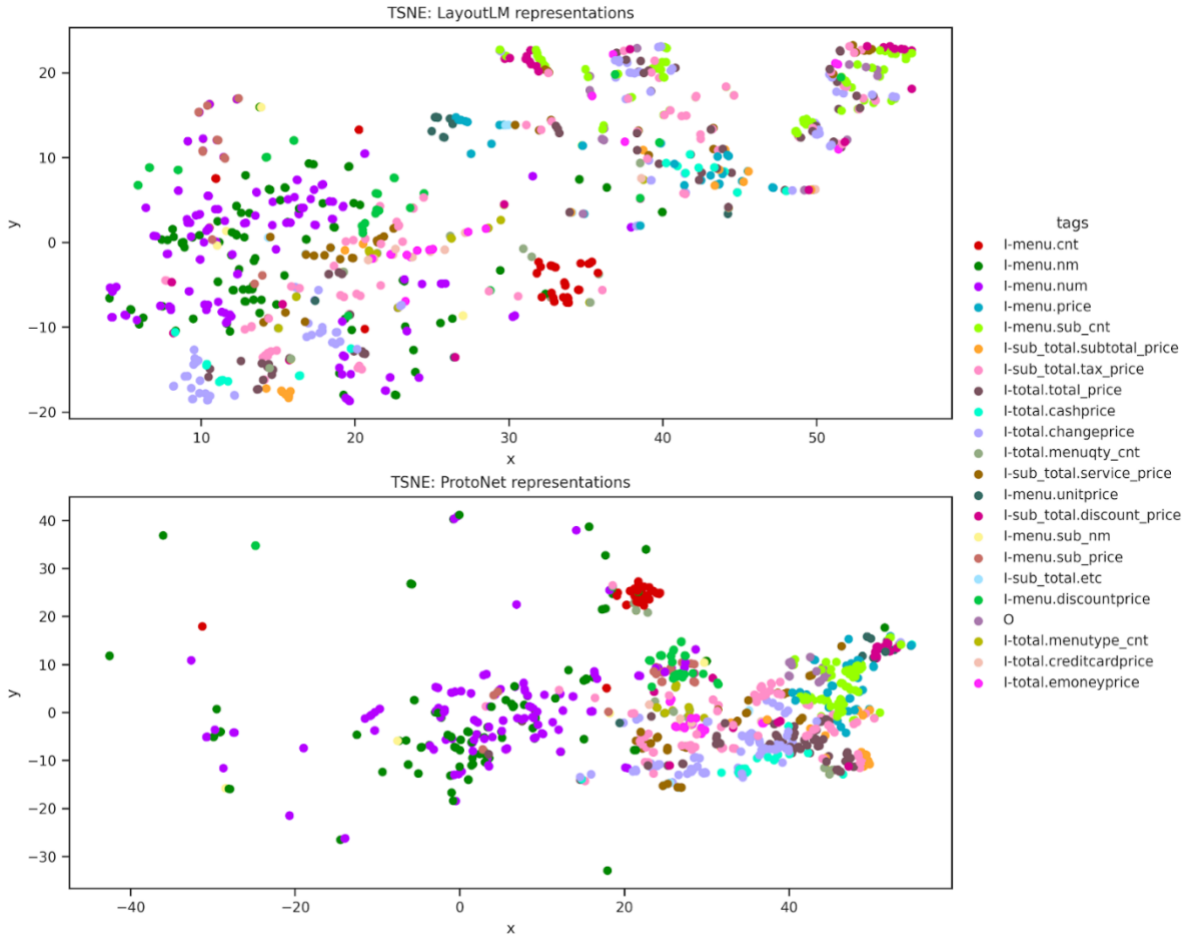

Figure 43 TSNE visualization of the ProtoNet-LayoutLM and its backend

Conclusion

We propose prototypical network model with large pretrained models as backend for few-shot entity recognition on visual documents. Our ProtoNet-LayoutLM model achieves an 81% accuracy on the CORD receipts dataset. Considering that only a small portion of the model parameters were trained, the approach is quite promising for information extraction in the few-shot regime when fully supervised datasets are not available, reducing significant cost and time.

In the future, we would like to improve our performance in one-shot learning and out of distribution settings. We intend to re-train our models end-to-end, fine-tuning the backends as well, and evaluate the performance. We would also like to explore zero-shot and weak-supervised learning scenarios.

REFERENCES

(n.d.). Retrieved from https://huggingface.co/datasets/katanaml/cord

Cui, L., Xu, Y., Lv, T., & Wei, F. (2021). Document ai: Benchmarks, models and applications. *arXiv preprint arXiv:2111.08609.*

Huang, Y., Lv, T., Cui, L., Lu, Y., & Wei., F. (2022). Layoutlmv3: Pre-training for document ai with unified text and image masking. *Proceedings of the 30th ACM International Conference on Multimedia*, (pp. 4083-4091).

Huang, Z., Chen, K., He, J., Bai, X., Karatzas, D., Lu, S., & Jawahar, C. V. (2019). Icdar2019 competition on scanned receipt ocr and information extraction. *International Conference on Document Analysis and Recognition (ICDAR)*, (pp. 1516-1520).

Park, S., Shin, S., Lee, B., Lee, J., Surh, J., Seo, M., & Lee, H. (2019). CORD: a consolidated receipt dataset for post-OCR parsing. *Workshop on Document Intelligence at NeurIPS 2019.*

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., . . . DeVito, Z. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32* (pp. 8024-8035). Curran Associates, Inc.

Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. *Advances in neural information processing systems, 30.*

Wang, J., Jin, L., & Ding, K. (2022). Lilt: A simple yet effective language-independent layout transformer for structured document understanding. *arXiv preprint arXiv:2202.13669.*