Selective Labeling: How to Radically Lower Data-Labeling Costs for Document Extraction Models

Anonymous ACL submission

Abstract

Building automatic extraction models for visually rich documents like invoices, receipts, bills, tax forms, etc. has received significant attention lately. A key bottleneck in developing extraction models for new document types is the cost of acquiring the several thousand high-quality labeled documents that are needed to train a model with acceptable accuracy. In this paper, we propose *selective labeling* as a solution to this problem. The key insight is to simplify the labeling task to provide "yes/no" labels for candidate extractions predicted by a model trained on partially-labeled documents. We combine this with a custom active learning strategy to find the predictions that the model is most uncertain about. We show through experiments on document types drawn from 3 different domains that selective labeling can reduce the cost of acquiring labeled data by $10\times$ while achieving negligible loss in accuracy.

1 Introduction

Visually rich documents such as invoices, receipts, paystubs, insurance statements, tax forms, etc. are pervasive in business workflows. The tedious and error-prone nature of these workflows has led to much recent research into machine learning methods for automatically extracting structured information from such documents (Lee et al., 2022; Garncarek et al., 2021; Xu et al., 2021; Tata et al., 2021). Given a target document type with an associated set of fields of interest, as well as a set of human-annotated training documents, these systems learn to automatically extract the values for these fields from documents with unseen layouts.

A critical hurdle in the development of high-quality extraction systems is the large cost of acquiring and annotating training documents belonging to the target types. The human annotators often require training not only on the use of the annotation tools but also on the definitions and semantics of the target document type. The annotation task can be tedious and cognitively taxing, requiring the annotator to identify and draw bounding boxes around dozens of target fields in each document. Not all the fields in the schema occur in all documents, leading to higher quality ground-truth annotations for the easier fields that occur frequently and lower quality annotations for infrequent fields, which are often missed.

This data efficiency requirement has not gone unnoticed in the research literature on this topic. Pre-training on large unlabeled document corpora (Xu et al., 2020, 2021) as well as applying transfer learning from an out-of-domain labeled corpus (Torrey and Shavlik, 2010; Nguyen et al., 2019) have both proven to be useful techniques in reducing the amount of training data required to get accurate models. However, even with these techniques, empirical evidence suggests that performing well on a new target document type still requires thousands of annotated documents, amounting to hundreds of hours of human labor (Zhang, 2021). Automating document-heavy business workflows in domains like procurement, banking, insurance, mortgage, etc. requires scaling to extraction models for hundreds of different document types.

The cost of acquiring high quality labeled data for hundreds of document types is prohibitively expensive and is currently a key bottleneck. We could apply active learning strategies to select a few but informative documents for human review (Settles, 2009), however the cost-reducing effect of this approach is limited, as it requires annotation of every candidate extraction span in every selected document, many of which may not be informative if they are repetitive, anomalous, or too easy for the extraction model to predict. In this paper, we propose a technique called *selective labeling* that reduces this cost by $10\times$. The key insight is to combine two ideas: First, we redefine and simplify the task performed by the human annotators – rather than labeling every target field in every document by drawing

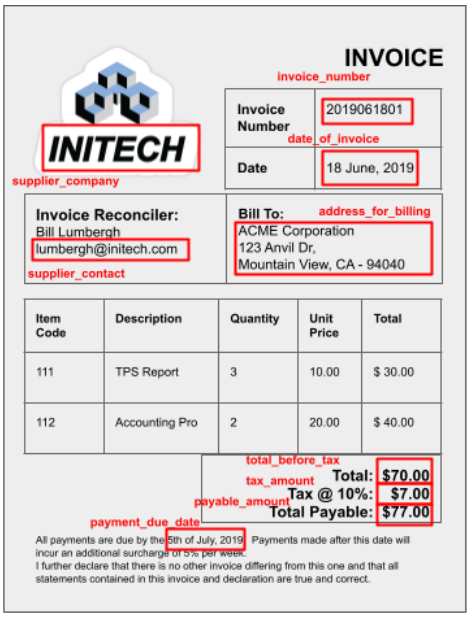


Figure 1: A classic annotation task. Even labeling 9 fields in this toy invoice imposes a heavy cognitive burden on the annotator, while real-world documents are significantly more complicated.

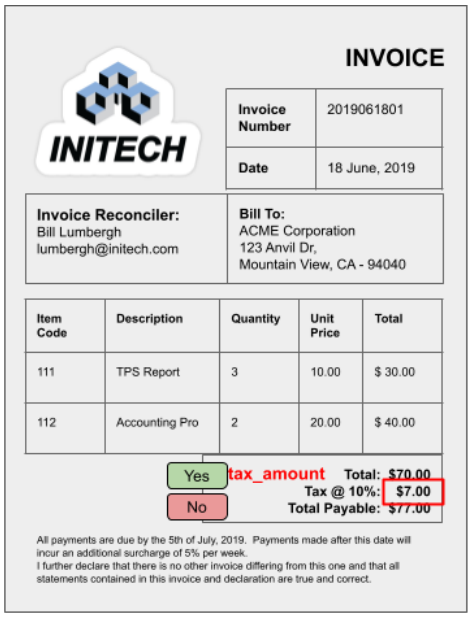


Figure 2: A "yes/no" annotation task. Presenting a proposed span and asking the annotator to accept or reject the label is simpler, quicker, and less prone to errors.

bounding boxes around their values, we ask them to simply verify whether a proposed bounding box is correct. This binary "yes/no" annotation task is faster and imposes a lighter cognitive burden on the annotator (Blog, 2020; Ganchev et al., 2007; Skeppstedt et al., 2017). Second, we adapt existing active learning strategies to select the examples (i.e., candidate extraction spans) that the model is most uncertain in each round to annotate.

We find that relying on a simple uncertainty metric, such as the distance between prediction scores and the middle point between the target labels (e.g., 0.5), is sufficient for selecting informative candidate extraction spans to annotate. We further propose new methods to increase diversity in the selection pool by reallocating the annotation budget to encourage selection of more infrequent fields. This is accomplished by calibrating the highly imbalanced prediction scores at the field level and limiting the number of candidates of each field to be reviewed in each document.

We interleave rounds of such human annotation with training a model that is capable of consuming partially-labeled documents. In combination, our proposed approach dramatically improves the efficiency of the annotation workflow for this extraction task. In fact, through experiments on document types drawn from multiple domains, we show that selective labeling allows us to build models with $10\times$ lower annotation cost while achieving nearly the same accuracy as a model trained on several thousand labeled documents.

2 Background

We first describe how a typical annotation task is set up to acquire labeled documents. We point out two major deficiencies with this approach before outlining an alternative that takes advantage of the characteristics of this domain. We then describe the assumptions underlying our approach.

2.1 Annotation Workflow

2.1.1 Classic Annotation Workflow

Given a document type for which we want to learn an extraction model, we begin by listing out the fields that we want to extract, along with human-readable descriptions, viz., "labeling instructions". We provide these instructions to human annotators and present them with various document images to label. The classic annotation task is to draw a bounding box around each instance of any of the target fields and label it with the corresponding field name (Figure 1). Typical document types like invoices and paystubs have dozens of fields, and each document may contain multiple pages.

The high cognitive burden of the classic annotation workflow leads to two major drawbacks. First, it makes training data collection extremely expensive. In one annotation task for paystub-like documents with 25 target fields, the average time to label

each document was about 6 minutes. Scaling this to hundreds of document types with thousands of documents each would be prohibitively expensive. Second, the resulting annotation quality is often quite poor. We have observed systematic errors such as missing labels for fields that occur infrequently in the documents or for instances that are in the bottom third of the page. To obtain acceptable training and test data quality, each document must be labeled multiple times, further exacerbating the annotation cost issue.

2.1.2 Proposed Annotation Workflow

We propose the following alternative to the classic annotation workflow:

(1) We speed up labeling throughput by simplifying the task: rather than drawing bounding boxes, we ask human annotators to accept or reject a candidate extraction. Figure 2 illustrates how much easier this "yes/no" task is compared to the classic one in Figure 1.

(2) We further cut down annotation cost by only labeling a subset of documents and only a subset of fields in each document.

(3) We use a model trained on partially-labeled documents to propose the candidate extraction spans for labeling. This allows us to interleave model training and labeling so that the model keeps improving as more labels are collected.

(4) We use a customized active learning strategy to identify the most useful labels to collect, viz., the candidate extraction spans about which the model is most uncertain. In successive labeling rounds, we focus our labeling budget on the fields that the model has not yet learned to extract well, such as the more infrequent ones.

In Section 5, we show empirical evidence that this improved workflow allows us to get to nearly the same quality as a model trained on 10k docs by spending an *order-of-magnitude less* on data-labeling. Note that naively switching the labeling task to the "yes/no" approach does not cut down the labeling cost – if we were to highlight every span that might potentially be an amount and present an "Is this the tax_amount?" question like in Figure 2, with the dozens of numbers that are typically present in an invoice, this workflow will be *much more* expensive than the classic one. A key insight we contribute is that a model trained on a modest amount of data can be used to determine a highly effective subset of "yes/no" questions to ask.

2.2 Assumptions

We make the following four assumptions about the problem setting: (1) We assume access to a pool of unlabeled documents. This is a natural assumption in any work on managing cost of acquiring labeled training data. (2) We assume the extraction model can be trained on partially labeled documents. (3) We assume the model can generate candidate spans for each field and a measure of uncertainty – this is used to decide the set of "yes/no" questions to present to the annotator. (4) The analysis in this paper uses empirical measurements for labeling tasks on documents with roughly 25 fields to model the costs of the traditional approach (6 minutes per document) and the proposed approach (10 seconds per "yes/no" question (Blog, 2020)). For more complex documents the difference in the two costs may be significantly higher.

Throughout this work, we use an extraction system similar to the architecture described in (Majumder et al., 2020). This architecture consists of two stages: candidate generation and candidate classification. In the first stage, we generate candidates for each field according to the type associated with that field. For example, the candidates generated for the *date of invoice* field would be the set of all dates in that invoice. The candidate generators for field types like dates, prices, numbers, addresses, etc. are built using off-the-shelf, domain agnostic, high-recall text annotation libraries. In the second stage, we score each candidate's likelihood of being the correct extraction span for the document and field it belongs to. This scoring is done using a neural network model trained as a binary classifier. The highest-scoring candidate for a given document and field is predicted as the extraction output for the document and field if it exceeds a certain field-specific threshold.

The ability to train on partially labeled documents is trivially true for this modeling approach since it employs a binary classifier trained on the labeled candidates. This should be relatively straightforward for sequence labeling approaches, such as (Xu et al., 2021), as well. Identifying a potential span in the document to present as a "yes/no" question to an annotator is an exercise in ranking the candidates for each field. We expect that sequence labeling approaches can be adapted to satisfy this requirement, e.g., by using beam search to decode the top few sequence labels. However, this is likely more complex than the aforementioned approach,

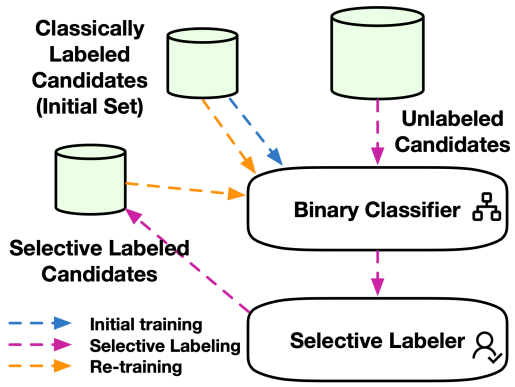


Figure 3: The model training pipeline starts by inital training (blue) the binary classifier using the small classically labeled dataset. We then selectively label (purple) a fixed number of candidates according to the budget, which are then used to re-train (orange) the model together with the initial dataset.

and we leave this as an exercise for future work.

3 Selective Labeling Methodology

We first provide an overview of the selective labeling framework before describing various uncertainty measures and ways to deal with the unique characteristics of this setting, such as varying difficulty for different fields.

3.1 Overview

Figure 3 provides a visual overview of our selective labeling workflow. Given a corpus of several thousand unlabeled documents belonging to the target document type, we begin by fully labeling a small randomly-sampled subset, say 50-250 docs, using the classic annotation workflow. We use this initial corpus to fine-tune a checkpoint originally trained on an out-of-domain corpus.

Our labeling workflow proceeds in rounds. In each round, we leverage the current model to select k candidate spans from the unlabeled set and have them reviewed by human annotators. The annotators answer a "yes/no" question (see Figure 2) either accepting or rejecting this proposed label. The newly labeled examples are merged into the training set and the model is retrained in each round. We repeat this iterative labeling-and-training procedure until we exhaust our annotation budget or reach our target F1 score.

The efficacy of this workflow clearly depends on the procedure we use to select candidate spans for human annotation. Based on the fundamental insight underlying much active learning literature, we select the candidates that the model is *most uncertain* about. In the remainder of this section, we describe how we adapt standard active learning strategies to a document extraction setting.

3.2 Measuring Uncertainty

There are a number of metrics we can use to quantify a model's prediction uncertainty (Lewis and Gale, 1994; Ko et al., 1995). In this work, we explored two types of uncertainty metrics.

Score distance. This method assigns a metric to each candidate based on the distance that the score is from some threshold (Li and Sethi, 2006). More formally, the uncertainty is $1-|score-threshold|$. For example, if the threshold is 0.5, this suggests that the model is most uncertain of its predictions of scores close to 0.5, in either direction.

Score variance. This method performs inference on a candidate multiple times with the dropout layer enabled and assigns the uncertainty metric as the variance of the scores (Gal and Ghahramani, 2016; Kirsch et al., 2019; Ostapuk et al., 2019). An alternative method trains multiple models independently from one another and assigns the uncertainty metric as the variance of the scores across all models (Seung et al., 1992). Note that empirically, we observed this yields near identical results as the dropout-based approach, so we only present findings for the latter.

3.2.1 Score Calibration

Our model's predicted scores tend to be uncalibrated, particularly in initial rounds and for infrequent fields due to training data scarcity. We calibrate scores in such a way that picking a candidate with a calibrated score of, say, 0.6 yields a 60% probability that it has a positive label (Guo et al., 2017). We compute calibration curves using the labeled training dataset by bucketing the candidates based on score. Note that we recompute the calibration curves for the new model after every round of selective labeling.

There are two interesting design choices we made in this process, both of which are made based on our knowledge of the score distribution. (1) The vast majority ($> 90\%$) of our candidates are negative and most of them have very low scores ($< 10^{-3}$), while the region of interest to us when calibrating the scores is the rest ($[10^{-3}, 1]$). In calculating bin edges, we exclude all candidates with scores that are smaller than a threshold (10^{-3}). All the scores below this threshold are placed in the first bin ($[0, 10^{-3})$). Since the vast majority of can-

didates get excluded by this filter, the remaining bins have a much higher resolution. (2) We use equal-frequency bins rather than equal-width bins because of the highly non-uniform distribution of scores, even within the score region of interest – in other words, each bin has roughly the same number of scores, except the first bin.

Once binned, calibration curves are computed for each field by interpolating between the curves *prevalence* (i.e., the proportion of candidates in each score bin that are positive) and the median scores for all the score bins.

By calibrating the scores, threshold selection becomes much more intuitive for the score-based uncertainty metric. For example, if we specify a threshold of 0.5, we expect that to mean we will select candidates for which the model has a 50% chance of classifying correctly *across all fields*.

3.3 Sampling Candidates

Once the uncertainty metric is calculated for each candidate in the unlabeled set, the next step is to select a subset of those candidates for human review. The most obvious method is to select the top-k candidates, thereby selecting the candidates for which the model is most uncertain. In practice, this can lead to sub-optimal results when the model finds many examples for which it is uncertain but may in fact be very similar to one another. The most common approach to break out of this trap is to introduce some notion of diversity in the sampling methodology (Gao et al., 2020; Ishii et al., 2002).

Combining Top-k and Random Sampling. A common method is to reallocate the k budget in each round so that a portion of that budget goes towards the top candidates by uncertainty (ensuring we get labels for the most uncertain candidates) and the remaining budget goes towards a random sample of candidates from the unlabeled set (ensuring that some amount of diversity is included in each round). One approach is to select the top-k' candidates by the uncertainty metric, where $k' < k$, and then randomly sample $k - k'$ candidates from the remaining unlabeled dataset. A second approach is to randomly sample k candidates from a pool of top-n candidates, where $n > k$. We found in practice that these two methods yield nearly identical results, so we only present findings for the first.

Capping Candidates for Each Document and Field. An important observation we make about the extraction problem is the following: While a given field typically has multiple candidates in every document, usually, at most one of these is positive and the rest are negative. For example, there are usually many dates in an invoice, and typically only one of them is the *date of invoice*. The uncertainty metrics we defined in Section 3.2 do not take into account this relationship between labels. We leverage this intuition to increase sample diversity by capping the number of candidates selected from the same document and field. After ordering the candidates by the chosen uncertainty metric, if we were to simply select the top-k candidates, we might end up selecting too many candidates for the same document and field. Instead, we select at most m candidates for each document and field, m being a tunable hyperparameter. This ensures that we spread the annotation budget over more documents and fields.

3.4 Automatically Inferring Negatives

After candidates have been selected and labeled, we merge the newly-labeled candidates into our training set. At this point, there is another opportunity to draw additional value from the unlabeled corpus by utilizing the structure of the extraction problem. The key insight here is that when a positive label is revealed via selective labeling, we can infer negative labels for some remaining candidates in the document.

If we assume that there is no more than one instance of a positive per field in a document then we can automatically infer that all of that field's remaining candidates in the document are negative. While for some fields it is possible that multiple instances of the same field appear on a document, we have found in practice that most fields only appear once in each document and applying this inference can collect more negative instances with useful contrastive knowledge.

4 Experiment Setup

To evaluate the performance of our proposed methods, we use datasets belonging to three different domains, summarized in Table 1. The number of fields varies across domains, e.g., the *Tax Forms* dataset has more than twice the fields as the *Retail Finance* dataset. We use hidden-label datasets instead of real unlabeled datasets and simulate the labeling procedure by revealing the labels of the candidates from the hidden-label datasets.

Recall from Section 2 that we employ two annotation methods: the classic annotation method (6

Domain	# Fields	Splits	# Docs	# Candidates
Supply Chain	18	Initial-50	50	11.8K
		Initial-100	100	24.5K
		Initial-250	250	58.7K
		Test	5,019	1.2M
		Hidden-label	10,000	2.4M
Retail Finance	11	Initial-100	100	76.0K
		Test	849	1.2M
		Hidden-label	4,000	5.6M
Tax Forms	24	Initial-100	100	13.4K
		Test	1,498	1.0M
		Hidden-label	7,500	5.1M

Table 1: Statistics of datasets in three domains.

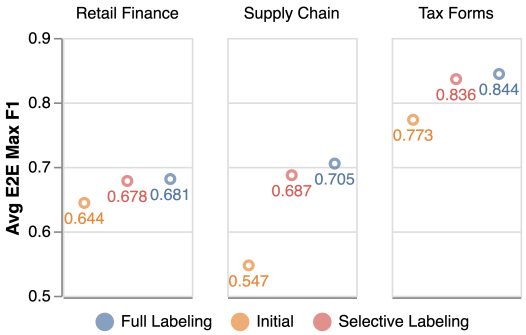


Figure 4: Best performing Selective Labeling as compared to Initial which is trained on just 100 documents and Full Labeling in which the hidden-label dataset (used in Selective Labeling) is fully used in training.

minutes per document), which is always applied to the initial training set, and the proposed "yes/no" method (10 seconds per candidate), which is applied during the selective labeling procedure on the unlabeled dataset. To explore how the size of the initial labeled dataset impacts our methods, we create three initial splits for the *Supply Chain* domain with 50, 100, and 250 documents.

In all of our experiments, we split the train set into 80-20 training-validation sets. The validation set is used to pick the best model by AUC-ROC, and we use the test split to report the performance metrics. We train using the Rectified Adam (Liu et al., 2020) optimizer with a learning rate of 0.001 for 25 epochs and set the dropout rate to 0.1 and batch size to 128. We also measure AUC-ROC on the validation set to decide whether to trigger early stopping after 3 epochs of no improvement. Finally, we evaluate our methods by measuring the overall extraction system's performance on the test set using the maximum F1 averaged across all fields, denoted as "Average E2E Max F1" in (Majumder et al., 2020). Every reported F1 score is further averaged over 10 independent runs to account for variability. After applying grid search to tune the hyperparameters, we specify $k' = 0.9k$ and sample at most $m = 1$ candidates for each document and field. The binary classifier has 330k parameters and each set of experiments trained within 4 hours on a NVIDIA Tesla P100 GPU.

5 Results

In this section, we present the overall performance of our best selective labeling strategy on three domains, a comparison of the different selection metrics, sampling methodologies, and how the number of rounds of selective labeling affects performance. We perform an ablation study to understand the effectiveness of our proposed diversity techniques, and finally demonstrate how performance varies with the size of the initial labeled dataset.

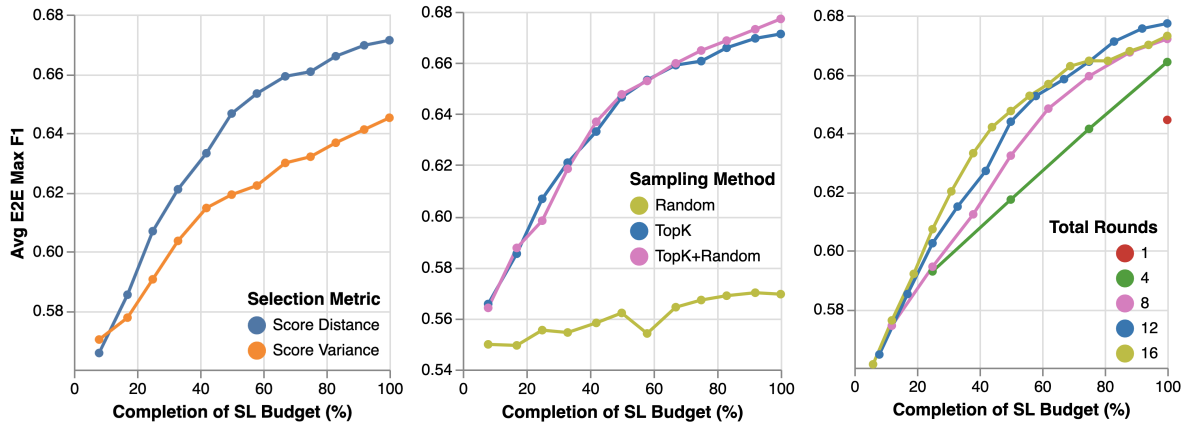
5.1 Best Performance on Different Domains

We train three initial models on a randomly sampled and labeled set of 100 documents for each domain. For example, as shown in Figure 4, the initial model for the *Supply Chain* domain achieves 0.547 F1 on the test dataset. We fine-tune the initial model on a fully labeled 10k document dataset (i.e., the hidden-label set from Table 1, in which for the purposes of this analysis we use its true labels), resulting in an F1 score of 0.705. The performance gap between these two models is thus 0.158.

Starting from the same initial model, we apply our best selective labeling strategy (which we discuss in the following sections) to reveal the labels from a subset of candidates that comprises only 10% of the annotation cost of fully labeling the hidden-label dataset. For the *Supply Chain* domain, this achieves an F1 score of 0.687, which closes the performance gap by 89%. Similarly, we close the gap by 88% and 92% for the *Retail Finance* and *Tax Forms* domains, respectively. This demonstrates that our method can dramatically decrease the annotation cost without sacrificing much performance.

5.2 Selection Metrics

In Figure 5a we plot per-round performance of two selection metrics in the *Supply Chain* domain given the same set of documents and annotation budget (i.e, 10% cost) and using the top-k sampling methodology. We observe that not only is computing score distances as the uncertainty indicator much more computationally efficient than variance-based metrics (10× faster), but it also significantly outperforms the latter as well. As we exhaust the budget over time, the advantage of score distance becomes more obvious.



(a) (b) (c)

Figure 5: Performance comparisons between (a) selection metrics, (b) sampling approaches, and (c) the rate at which we exhaust the budget through different number of rounds of selective labeling. The x-axis denotes the percentage of the total selective labeling budget consumed.

5.3 Sampling Methodology

Figure 5b compares performance across different sampling methodologies. As one might expect, pure random sampling is far worse than any other approach – we believe the initial model is confident in predicting a large quantity of candidates (especially the negatives), and randomly sampling from them does not obtain much useful knowledge.

The top-k strategies produce much more impressive results. Furthermore, we observe in later rounds that injecting some diversity via randomness achieves slightly better performance than the vanilla top-k approach. We believe this mimics the aggregation of exploitation (top-k) and exploration (random) processes, proven to be beneficial in reinforcement learning applications (Ishii et al., 2002). This also confirms our suspicion that top-k alone can lead us into selecting many uncertain examples which are in fact very similar to one another.

5.4 Multi-round Setting

In Figure 5c, we compare 5 learning curves, each of which denotes selecting the same number of candidates in total (10% annotation cost) over a different number of rounds. For example, the 16-round experiment selects $\frac{1}{16}$ of the total budget in each round, while the 1-round experiment utilizes the entire budget in a single round.

As we increase the total number of rounds, the model tends to yield better extraction performance until it peaks at about 12 rounds. This finer-grained strategy usually performs better than coarser ones but the gains become marginal at a higher number of rounds. Interestingly, we find that using up just half the budget in the first 8 rounds of a 16-round experiment achieves slightly better performance than exhausting the entire budget in the 1-round experiment. This comparison underscores the importance of employing a multi-round approach.

Models	Avg E2E Max F1 (std.)	Δ
SL	0.671 (0.006)	-
SL+CS	0.679 (0.005)	+1.2%
SL+CC	0.675 (0.005)	+0.6%
SL+AIN	0.683 (0.009)	+1.8%
SL+CS+CC+AIN	0.687 (0.005)	+2.1%

Table 2: Ablation Study. SL denotes selective labeling utilizing the top-k sampling and score distance metric. CS, CC, and AIN represent calibrating scores, capping candidates and automatically inferring negatives.

5.5 Ablation Study

Table 2 presents an ablation study to understand the impact of different diversity strategies. SL represents a 12-round selective labeling method using top-k sampling on the score distance metric. We separately add one feature at a time to test the effectiveness of calibrating scores (CS), automatically inferring negatives (AIN) and capping candidates (CC). Results show that every feature improves the model, but we achieve the largest improvement when applying all features in SL+CS+CC+AIN. It is reasonable to conclude that increasing diversity intelligently helps us select more useful candidates than relying on the uncertainty metric alone.

5.6 Initial Labeled Dataset Size

Given the dependence of the selective labeling method on an initially labeled small dataset, it is imperative that we evaluate how the approach is affected by the number of documents in this initial

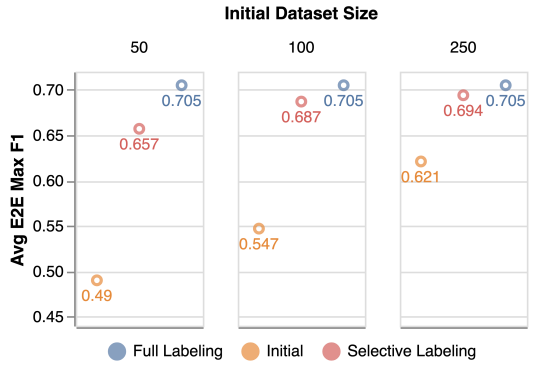


Figure 6: Comparison among three initial dataset sizes in the *Supply Chain* domain. We present the same three approaches as in Figure 4: Initial is trained on the initial dataset alone, Selective Labeling selects the equivalent of 10% annotation cost in candidates, and Full Labeling fine-tunes from the initial model on the full hidden-label data.

dataset. We experiment with initial datasets of 50, 100, and 250 documents in the *Supply Chain* domain using our best selective labeling strategy and a budget equivalent of 10% cost of annotating the "unlabeled" dataset.

Figure 6 indicates that the size of the initial dataset greatly impacts the performance of the model trained solely on those initial training sets, but has starkly less of an impact once we apply selective labeling. We close the performance gap by 77%, 89%, and 87%, for initial dataset sizes of 50, 100, and 250, respectively. We can conclude that selective labeling is capable of finding useful candidates to significantly improve the model performance even at a cost of only 10% of the annotation budget. And it is not surprising that the selective labeling gains may suffer when the initial dataset is too small (e.g. 50).

6 Related Work

Form Extraction. There have been numerous recent studies on information extraction for form-like documents. Existing approaches either individually categorize every text span in the document (Majumder et al., 2020) or formulate the task into a sequence modeling problem (Aggarwal et al., 2020; Lee et al., 2022; Garncarek et al., 2021; Xu et al., 2021) and encode texts, layouts, and visual patterns into feature space. While these approaches produce state-of-the-art extraction systems, they require large amounts of labeled training data to do so. In our work, we do not propose a new model architecture but instead, focus on the cost of acquiring labeled data for such extraction models.

Active Learning. We refer to (Settles, 2009) for an extensive review of the literature. In our work, we are interested in a pool-based selection strategy that assumes a large unlabeled set to select samples from and request for human annotation. Two popular approaches for requesting annotation are (1) uncertainty-based selection (Lewis and Gale, 1994) which can measure the uncertainty based on entropy (Ko et al., 1995), least confidence (Culotta and McCallum, 2005), or maximum margin (Boser et al., 1992); and (2) committee-based selection (Seung et al., 1992), which select instances based on disagreement upon multiple predictions (Gal and Ghahramani, 2016; Kirsch et al., 2019). Methods that are only concerned with uncertainty might introduce redundancy or skew the model towards that particular area of the distribution. Researchers seek to increase the diversity by forcing the selection to cover a more representative set of examples (Yang et al., 2017; Yin et al., 2017; Sener and Savarese, 2018) or incorporating discriminative learning to make the labeled set and the unlabeled pool indistinguishable (Gissin and Shalev-Shwartz, 2019).

To the best of our knowledge, we are the first to customize active learning strategies to reduce the annotation cost in the form-like document extraction task. In our selective labeling experiments, we explore a variety of informativeness-based selection strategies due to their simplicity and promising performance. We also explore introducing diversity by reallocating a portion of the labeling budget for random sampling as well as through proposing task-aware methods, such as automatic negative inference and capping candidates.

7 Conclusion

We have presented a new approach to acquire labeled data for form extraction tasks that reduces the annotation cost by $10\times$ as compared to fully labeling a large corpus, without sacrificing much extraction performance. The key insight is to transform the annotation task into a "yes/no" task and leverage a model type that can be trained on partially labeled documents in a multi-round active learning setting. We proposed novel techniques that take advantage of the characteristics of the problem to further improve extraction performance in the context of our selective labeling strategy. Thus, our approach has the potential to overcome the bottleneck of obtaining large amounts of high-quality training data for hundreds of document types.

References

Milan Aggarwal, Hiresh Gupta, Mausoom Sarkar, and Balaji Krishnamurthy. 2020. Form2Seq : A framework for higher-order form structure extraction. In *Proceedings of theConference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3830–3840.

CloudResearch Blog. 2020. A Simple Formula for Predicting the time to complete a study on mechanical Turk.

Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152.

Aron Culotta and Andrew McCallum. 2005. Reducing labeling effort for structured prediction tasks. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, volume 5, pages 746–751.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1050–1059. PMLR.

Kuzman Ganchev, Fernando Pereira, Mark Mandel, Steven Carroll, and Peter White. 2007. Semi-automated named entity annotation. In *Proceedings of the Linguistic Annotation Workshop*, pages 53–56.

Mingfei Gao, Zizhao Zhang, Guo Yu, Sercan Ö. Arık, Larry S. Davis, and Tomas Pfister. 2020. Consistency-based semi-supervised active learning: Towards minimizing labeling cost. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 510–526.

Łukasz Garncarek, Rafał Powalski, Tomasz Stanisławek, Bartosz Topolski, Piotr Halama, Michał Turski, and Filip Graliński. 2021. LAMBERT: Layout-aware language modeling for information extraction. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, pages 532–547. Springer.

Daniel Gissin and Shai Shalev-Shwartz. 2019. Discriminative active learning. *arXiv preprint arXiv:1907.06347*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.

Shin Ishii, Wako Yoshida, and Junichiro Yoshimoto. 2002. Control of exploitation–exploration meta-parameter in reinforcement learning. *Neural Networks*, 15(4-6):665–687.

Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. 2019. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in Neural Information Processing Systems*, 32.

Chun-Wa Ko, Jon Lee, and Maurice Queyranne. 1995. An exact algorithm for maximum entropy sampling. *Operations Research*, 43(4):684–691.

Chen-Yu Lee, Chun-Liang Li, Timothy Dozat, Vincent Perot, Guolong Su, Nan Hua, Joshua Ainslie, Renshen Wang, Yasuhisa Fujii, and Tomas Pfister. 2022. FormNet: Structural encoding beyond sequential modeling in form document information extraction. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

David D Lewis and William A Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12. Springer.

Mingkun Li and Ishwar K Sethi. 2006. Confidence-based active learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1251–1261.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020. On the variance of the adaptive learning rate and beyond. In *Proceedings of the International Conference on Learning Representation (ICLR)*.

Bodhisattwa Majumder, Navneet Potti, Sandeep Tata, James B Wendt, Qi Zhao, and Marc Najork. 2020. Representation learning for information extraction from form-like documents. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6495–6504.

Minh-Tien Nguyen, Viet-Anh Phan, Le Thai Linh, Nguyen Hong Son, Le Tien Dung, Miku Hirano, and Hajime Hotta. 2019. Transfer learning for information extraction with limited data. In *Proceedings of the International Conference of the Pacific Association for Computational Linguistics*, pages 469–482. Springer.

Natalia Ostapuk, Jie Yang, and Philippe Cudré-Mauroux. 2019. Activelink: deep active learning for link prediction in knowledge graphs. In *The World Wide Web Conference*, pages 1398–1408.

Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Burr Settles. 2009. *Active learning literature survey*. Computer Sciences Technical Report 1648. University of Wisconsin-Madison.

H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. 1992. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 287–294.

Maria Skeppstedt, Carita Paradis, and Andreas Kerren. 2017. PAL, a tool for pre-annotation and active learning. *Journal for Language Technology and Computational Linguistics*, 31(1):91–110.

Sandeep Tata, Navneet Potti, James B Wendt, Lauro Beltrao Costa, Marc Najork, and Beliz Gunel. 2021. Glean: Structured extractions from templatic documents. *Proceedings of the International Conference on Very Large Databases*, 14(6):997–1005.

Lisa Torrey and Jude Shavlik. 2010. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, pages 242–264. IGI global.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2021. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2579–2591.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. LayoutLM: Pre-training of text and layout for document image understanding. In *Proceedings of the International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1192–1200. ACM.

Lin Yang, Yizhe Zhang, Jianxu Chen, Siyuan Zhang, and Danny Z Chen. 2017. Suggestive annotation: A deep active learning framework for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 399–407. Springer.

Changchang Yin, Buyue Qian, Shilei Cao, Xiaoyu Li, Jishang Wei, Qinghua Zheng, and Ian Davidson. 2017. Deep similarity-based batch mode active learning with exploration-exploitation. In *Proceedings of the International Conference on Data Mining (ICDM)*, pages 575–584. IEEE.

Cha Zhang. 2021. Visual document intelligence in the wild. Document Intelligence Workshop at KDD.