Data-Efficient Information Extraction from Form-Like Documents

Beliz Gunel*
Stanford University
bgunel@stanford.edu

Navneet Potti
Google
navsan@google.com

Sandeep Tata
Google
tata@google.com

James B. Wendt
Google
jwendt@google.com

Marc Najork
Google
najork@google.com

Jing Xie
Google
lucyxie@google.com

ABSTRACT

Automating information extraction from form-like documents *at scale* is a pressing need due to its potential impact on automating business workflows across many industries like financial services, insurance, and healthcare. The key challenge is that form-like documents in these business workflows can be laid out in virtually infinitely many ways; hence, a good solution to this problem should generalize to documents with *unseen* layouts and languages. A solution to this problem requires a holistic understanding of both the textual segments and the visual cues within a document, which is non-trivial. While the natural language processing and computer vision communities are starting to tackle this problem, there has not been much focus on (1) data-efficiency, and (2) ability to generalize across different document types and languages.

In this paper, we show that when we have only a small number of labeled documents for training (~50), a straightforward transfer learning approach from a considerably structurally-different larger labeled corpus yields up to a 27 F1 point improvement over simply training on the small corpus in the target domain. We improve on this with a *simple multi-domain transfer learning approach*, that is currently in production use, and show that this yields up to a *further* 8 F1 point improvement. We make the case that data efficiency is critical to enable information extraction systems to scale to handle hundreds of different document-types, and learning good representations is critical to accomplishing this.

ACM Reference Format:
Beliz Gunel, Navneet Potti, Sandeep Tata, James B. Wendt, Marc Najork, and Jing Xie. 2021. Data-Efficient Information Extraction from Form-Like Documents. In *Proceedings of Document Intelligence Workshop (KDD '21).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

*Work done during Google AI research internship, correspondence to bgunel@stanford.edu.

KDD '21, August 14–18, 2021, Singapore

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Given a target set of fields for a particular document type, say, *invoice date* and *total amount* for invoices, along with a small set of manually-labeled documents, the task at hand is to learn to automatically extract these fields from documents with *unseen* layouts and languages. Note that even within the documents of the same *type* and language, say English invoices, same pieces of information may be described in entirely different ways, as each vendor often has their own layout structure. We will refer to this layout structure as *template* for the rest of the paper. Moreover, this information extraction task requires understanding both the textual segments and the visual cues within a document as it aims to generalize to unseen templates across different document types and languages. Hence, the traditional information extraction techniques from webpages, most of which do integrate visual layout information [2–4, 15, 16], do not suffice here.

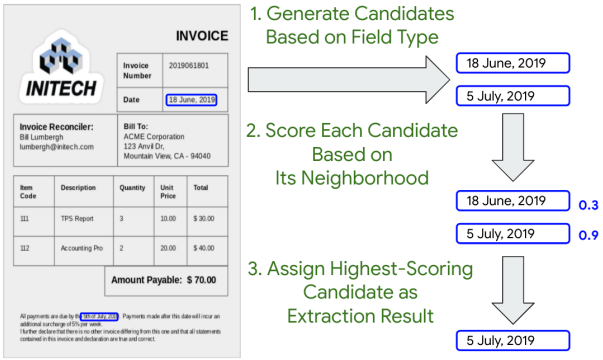

Figure 1: Given a document image and a target schema as inputs, we perform Optical Character Recognition (OCR) on the document image, generate candidates using candidate generators that leverage the existing domain knowledge of working with structured documents, score these generated candidates using a representation learning-based ML model that is described in Section 2, and assign the best candidates to the target fields to produce the final extraction result.

There have been several multi-modal ML-based proposals to tackle this task such as BERTGrid [5]. BERTGrid extends Katti et al. [8]'s work that previously proposed inputting documents as 2D grids of text tokens to fully convolutional encoder-decoder networks by incorporating pre-trained BERT [6] text embeddings

into the 2D grid representation. Another line of work proposes extending language model pre-training approaches to include the document layout information including [7, 14]. Although these approaches are promising, training or pre-training time of their pipelines are not only compute- and data-intensive, but also need to be *re-done from scratch* for competitive extraction performance while working with a new considerably structurally different document type or a different language. We would like to point out that in order to fully automate this task, we need to tackle 100s of document types – and the main cost is data acquisition and labeling for every new language or document type. If we can get to same extraction performance with 10x less data, we effectively cut the cost of developing new extraction models by 10x.

In our work, we borrow the basic architecture from Glean [11, 13], an information extraction system that uses a factored approach. Glean decouples the task into three stages – candidate generation, ML-based scoring, and assigning – as described in Figure 1. Glean's design decision of leveraging candidate generators built using standard entity extraction libraries significantly narrows down the search problem for its ML-based scorer model. This way, Glean's ML-based scorer model can fully focus on (1) learning a representation for an extraction candidate that captures its spatial neighborhood on the page, as well as (2) learning the semantics of the target field. We summarize the design of the extraction system and its ML-based scorer model that effectively leverages representation learning [1] ideas in Section 2.

In this paper, we explore whether it is possible to transfer knowledge over to (1) a considerably structurally-different document type in the same language or to (2) same document type in a different language in the context of Glean, when we have several orders of magnitude less labeled documents in the new document type or language. It is, again, crucial to note that this is of great practical importance, as it is often prohibitively expensive to gather and label datasets of large size for every new document type or language.

We build on the hypothesis that form-like documents share a *visual design language* and that the representation learning approach of Glean naturally enables multi-domain training and fine-tuning across different document types and different languages, by its design. In other words, we postulate that a representation learning approach that exploits that *shared visual design language across form-like documents* is precisely why we can effectively transfer knowledge across considerably different domains. The core idea we follow is that we first focus on learning a good encoder for the extraction candidates that understands the spatial relationships and semantics of form-like documents, and then we fine-tune the learned candidate encoder and the field-specific encodings on the new document type or language of interest.

We show that our *very simple multi-domain transfer learning approach* –that combines extraction candidates from both *source* and *target* domains, and use a common vocabulary across both these domains to train the scorer model followed by fine-tuning the model on the target domain – enables remarkable data-efficient generalization both from English Invoices to considerably structurally-different new document type Paystubs, and from English Invoices to French Invoices. Our proposed approach consistently improves over both *training from scratch* and simple *transfer learning* baselines up to 1k labeled documents. The value of our proposed approach is particularly impressive in the low data regimes. Specifically, we improve on the training from scratch baseline by up to 35 F1 points, and on the simple transfer learning baseline by up to 8 F1 points for the 50 labeled document case while generalizing to a new document type. Similarly, we improve on the training from scratch baseline by up to 23 F1 points, and on the simple transfer learning baseline by up to 7 F1 points for the 10 labeled document in the target domain case while generalizing to a new language.

2 EXTRACTION SYSTEM OVERVIEW

We build on Glean [13] extraction system that is described in Figure 1. Glean takes in a document image and a target schema –that includes target fields and their corresponding types– as inputs, and performs Optical Character Recognition (OCR). As an example, target schema for *Invoice* document type could include target fields such as *invoice date* of type *date* and total amount of type *price*. Glean supports numerous field types including *integer*, *numeric*, *alphanumeric*, and *currency*, *address*, *phone_number*, *url*, and other common entity types. Glean leverages an existing library of entity detectors used in Google's Knowledge Graph and are available through a Cloud API [1] for all the types described. Open-source entity detection libraries can be used for common types like names, dates, currency amounts, numbers, addresses, URLs, etc. [2] The candidate generators are designed to be high-recall – they identify *every* text span in the document that is likely to be of their type.

Once extraction candidates have been generated, an ML-based Scorer is used to assign a score for each (*field*, *candidate*) pair that estimates the likelihood that the given extraction candidate is the right extraction value for that field. Multiple fields in the target schema may belong to the same type, say *invoice date* and *due date*, and may therefore share the same set of extraction candidates. An extraction candidate is represented by the text span identified by the candidate generator along with context such as text in its immediate neighborhood to provide the ML-based scorer model with additional relevant features. Finally, candidates for a target field and document are independently scored, and highest scoring candidate for the field is assigned as the final extraction value. Note that, (1) additional business logic specific to a document type can be specified at the assignment stage such as including constraints like *invoice date* must precede *due date* chronologically, (2) precision of the overall extraction system can be adjusted by imposing a minimum score threshold for each field in the target schema.

A high-level abstraction for the ML-based Scorer is shown in Figure 2. Modeling specifics of the Scorer architecture is not the focus of this paper, and we refer the reader to Majumder et al. [11] for details. In this section, we explain the ML-based Scorer model architecture at a high-level in order for us to qualify *why* the representation learning inspired modeling choices naturally enable multi-domain training and fine-tuning across different document types and different languages. The features of each extraction candidate supplied to the model are its neighboring words and their relative positions, as visualized in Figure 2. Note that we exclude the candidate's value from the set of features in order to avoid biasing the model towards the distribution of values seen during

[1] https://developers.google.com/knowledge-graph
[2] https://cloud.google.com/natural-language/docs/reference/rest/v1/Entity

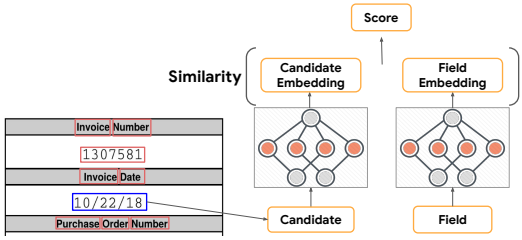

Figure 2: A candidate's score is based on the similarity between its embedding and a field embedding. A date extraction candidate "10/22/18" is shown in blue, along with its neighboring tokens, shown in orange.

training, which may not be representative of the entire domain at test time. Model learns a dense representation for each extraction candidate using a simple self-attention based architecture. Separately, in the same embedding space, it learns dense representations for each field in the target schema that capture the semantics of the fields. Based on these learned candidate and field representations, each extraction candidate is scored based on the similarity to its corresponding field embedding. The model is trained as a binary classifier using cross-entropy loss, where the target labels are obtained by comparing the candidate to the ground truth. Please refer to Tata et al. [13] for how the training data consisting of positive and negative extraction candidates are generated, along with the design decisions to address the data management challenges that arise due to the nature of the problem.

3 DATA EFFICIENCY

3.1 Experimental Setup

Datasets and Evaluation Metrics Dataset statistics are summarized in Table 1. During data curation, we ensured that no two documents in the corpus share the same template as we aim to learn to extract from *any* template. All of these datasets are proprietary, and hence, are unfortunately not available publicly. Our primary metric is the end-to-end extraction performance measured using Max F1 in the precision-recall curve, which we refer to as the *F1 score*. We always report macro-average F1 score across fields in the target schema. Note that F1 score is affected by the performance of all the parts within the pipeline including the quality of OCR engine and recall of the candidate generators. All experiments below use a batch size of 256, the Rectified Adam optimizer [9] with a learning rate of 0.001. These hyperparameters were found to be optimal using a grid-search. We train the models using an 80-20 train-validation split for up to 25 epochs in each stage, and pick the checkpoint with the best validation ROC AUC for the scorer model, which typically occurs in fewer than 10 epochs. Experiments on generalization to new language all use a vocabulary consisting of the 2k most frequent tokens occurring in English and French Invoice documents. Experiments on generalization to new document type all use a vocabulary consisting of the 4k most frequent occurring in Paystub and English Invoice documents. All experiments were repeated using 10 different seeded random initializations of the model, and we report the median performance on a fixed, hold-out test set of target domain documents along with error bars that show the variance across these runs.

	# Docs	# Fields	Usage
English Invoices	20k	12	Train
French Invoices Corpus1	5k	12	Train
French Invoices Corpus2	400	12	Test
Paystubs Corpus1	10k	19	Train
Paystubs Corpus2	1.5k	19	Test

Table 1: Dataset statistics. # Docs refer to the number of documents in the corpora, and # Fields refer to the number of fields in the schema for the given document type.

3.2 Multi-Domain Transfer Learning

Our proposed *multi-domain transfer learning* approach is a natural extension of the representation learning ideas used in Glean. For the remainder of this paper, we will refer to the *document type* or *language* that we already have enough labeled examples for as *source domain*; and the *document type* or language that we would like to generalize to but not have enough labeled examples as *target domain*. Note that this is of great practical importance, as it is often simply not feasible to gather and label datasets of large size for every new document type or language.

We build on the hypothesis of Glean that form-like documents share a visual design language and that the candidate encoder within the ML-based Scorer we use can effectively learn to represent the domain-agnostic spatial relationships – between the candidate and its neighbors – that are critical to understand so that we can have a holistic understanding of form-like documents. Another key observation we make is that the candidate encoder learns embeddings for the neighboring tokens, and these word embeddings, while agnostic to the field, are likely to be specific to the domain used for training. Thus, at Stage 1, we (1) combine the extraction candidates from both the *source* and *target* domains, and (2) use a common vocabulary across both these domains while training Glean's ML-based Scorer model. Significant performance benefit we observe over simple *transfer learning* approaches stems from these two key decisions based on our domain-specific observations, as they make the learned candidate representations *more general*. Field-specific information continues to be encoded in the field embeddings within the same latent space. At Stage 2, we simply fine-tune both the candidate encodings and field embeddings on the *target* domain. Note that this framework can easily be extended to an arbitrary number of *source* and *target* domains.

Our results shown in Figure 3 indicate that our proposed *multi-domain transfer learning* approach enables remarkable data-efficient generalization both from English Invoices to considerably structurally-different new document type Paystubs, and from English Invoices to French Invoices – consistently improving over both *training from scratch* and simple *transfer learning* baselines up to 1k labeled documents. Summary of all compared methods is described in Figure 4. Note that *domain* refers to document type in the first described setting, and it refers to language in the second described setting. *training from scratch* baseline simply trains a model using only the labeled examples in the *target* domain. The simple *transfer learning*

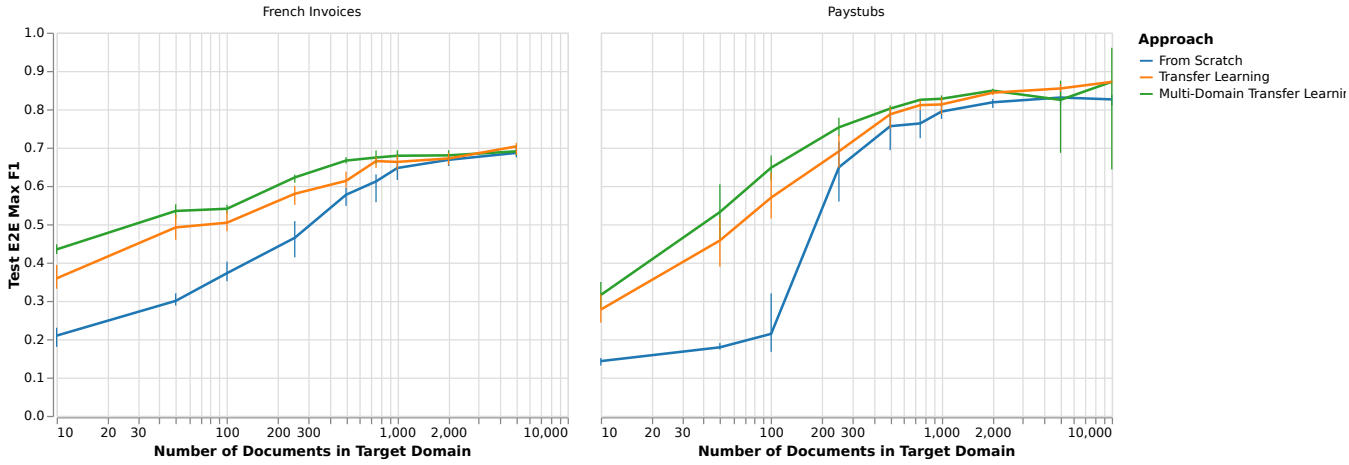

Figure 3: Both learning curves use the backbone architecture described in Section 2 and compare *train from scratch* and *transfer learning* baselines to our proposed *multi-domain transfer learning* approach. Left figure shows generalization ability from English Invoices to French Invoices, and right figure demonstrates generalization ability from English Invoices to Paystubs. All fields included in this analysis have above 80% candidate generation coverage and have at least 40 ground truth label in their corresponding test set. French Invoices document type have 12 and Paystubs have 19 fields in their target schemas. Both figures show median performance at different number of labeled documents for the *target domain* (new document type or language we are trying to generalize to) along with error bars that show variance across 10 different seeds. Proposed *multi-domain transfer learning* approach consistently improves on both baselines up to 1k labeled documents in the *target domain*, and the improvement is particularly significant in the low data regime. *Source domain* for both figures is English Invoices.

Approach	Initial Training Stage	Fine-tuning Stage
From Scratch	-	Target domain only
Transfer Learning	Source domain only	
Multi-domain Transfer Learning	Source & target domains	

Figure 4: All compared methods use the same backbone described in Section 2. Please refer to Figure 2 for high-level descriptions of Field Embedding and Candidate Encoder, and to [11] for specifics of the ML-based Scorer architecture.

baseline first trains the ML-based scorer model using the examples in the *source* domain and then fine-tunes the model in the *target* domain. Both learning curves shown in Figure 3 use the same backbone described in Section 2 and compare *train from scratch* and *transfer learning* baselines with our proposed *multi-domain transfer learning* approach. Left figure shows generalization ability from English Invoices to French Invoices, and right figure demonstrates generalization ability from English Invoices to Paystubs. All fields included in this analysis have above 80% candidate generation coverage and have at least 40 ground truth label in their corresponding test set. French Invoices document type have 12 and Paystubs have 19 fields in their target schemas. Both figures show median performance at different number of labeled documents for the *target domain* along with error bars. Our proposed *multi-domain transfer learning* approach consistently improves on both baselines up to 1k labeled documents in the *target domain*, and the improvement is particularly significant in the low data regime. *Source domain* for both cases is English Invoices.

The value of our proposed approach is particularly impressive in the low data regimes. Specifically, we improve on the training from scratch baseline by up to 35 F1 points, and on the simple transfer learning baseline by up to 8 F1 points for the 50 labeled document case while generalizing to a new document type. Similarly, we improve on the training from scratch baseline by up to 23 F1 points, and on the simple transfer learning baseline by up to 7 F1 points for the 10 labeled document case while generalizing to a new language. We also would like to point out that (1) source model training takes approximately 45 minutes, converging after 15-25 epochs, and fine-tuning on the target domain approximately takes couple minutes, converging after 1-2 epochs on a single GPU, in contrast to the pre-training based approaches such as BERTGrid model training that takes approximately 1090 minutes converging after 20 epochs based on our own implementation, (2) our proposed multi-domain transfer learning approach is currently in production use.

4 RELATED WORK

Katti et al. [8] propose inputting documents as 2D grids of text tokens to fully convolutional encoder-decoder networks. Denk and Reisswig [5] incorporate pretrained BERT text embeddings into that 2D grid representation. Xu et al. [14] propose integrating 2D position embeddings and image embeddings, produced with a Faster R-CNN [12] model, into the backbone structure of a BERT language model [6] and using a masked visual-language loss during pre-training. Similarly, Garncarek et al. [7] propose integrating the 2D layout information into the backbone structure of both BERT and RoBERTa [10], where they construct layout embeddings using a graph neural network using a heuristically constructed document

graph. In contrast to these pre-training based approaches, Glean extraction system that we build on (1) requires several orders of magnitude less labeled training data (2) an order of magnitude less training and inference time for the parts of the extraction system that use ML, without sacrificing the generalization ability to new document types and languages.

5 DISCUSSION

We argued that data-efficiency will be immensely critical as the information extraction systems in production will increasingly need to perform well across *more* document types, *more* languages, and potentially on private customer data – ideally without sacrificing the generalization ability and the training and inference time of the parts of the extraction system that use ML. We hope that our preliminary results will help start the discussion on the *importance of data-efficiency* while building the next generation information extraction systems tailored to form-like documents for production use. We believe that next big step will be to decrease the labeled document need from ∼1k to ∼100 for each new (n+1)th document type or language we would like to generalize to.

REFERENCES

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 8 (2013), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

[2] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. 2003. Extracting Content Structure for Web Pages Based on Visual Representation. In *Web Technologies and Applications, APWeb*. 406–417. https://doi.org/10.1007/3-540-36901-5_42

[3] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. 2004. Block-based web search. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 456–463. https://doi.org/10.1145/1008992.1009070

[4] Gobinda G. Chowdhury. 1999. Template Mining for Information Extraction from Digital Documents. *Library Trends* 48, 1 (1999), 182–208. https://www.ideals.illinois.edu/handle/2142/8258

[5] Timo I. Denk and Christian Reisswig. 2019. BERTgrid: Contextualized Embedding for 2D Document Representation and Understanding. *ArXiv* abs/1909.04948 (2019). http://arxiv.org/abs/1909.04948

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186. https://www.aclweb.org/anthology/N19-1423/

[7] Lukasz Garncarek, Rafal Powalski, Tomasz Stanislawek, Bartosz Topolski, Piotr Halama, and Filip Gralinski. 2020. LAMBERT: Layout-Aware language Modeling using BERT for information extraction. *ArXiv* abs/2002.08087 (2020). https://arxiv.org/abs/2002.08087

[8] Anoop R. Katti, Christian Reisswig, Cordula Guder, Sebastian Brarda, Steffen Bickel, Johannes Höhne, and Jean Baptiste Faddoul. 2018. Chargrid: Towards Understanding 2D Documents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4459–4469. https://www.aclweb.org/anthology/D18-1476/

[9] 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 9th International Conference on Learning Representations*. https://openreview.net/forum?id=rkgz2aEKDr

[10] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv* abs/1907.11692 (2019). https://arxiv.org/abs/1907.11692

[11] Bodhisattwa Prasad 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 58th Annual Meeting of the Association for Computational Linguistics*. 6495–6504. https://doi.org/10.18653/v1/2020.acl-main.580

[12] 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, 6 (2015), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

[13] Sandeep Tata, Navneet Potti, James B. Wendt, Lauro Beltrao Costa, Marc Najork, and Beliz Gunel. 2021. Glean: Structured Extractions from Templatic Documents. In *Proceedings of the VLDB Endowment*. 997–1005. https://doi.org/10.14778/3447689.3447703

[14] 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 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1192–1200. https://doi.org/10.1145/3394486.3403172

[15] Shipeng Yu, Deng Cai, Ji-Rong Wen, and Wei-Ying Ma. 2003. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *Proceedings of the 12th International Conference on World Wide Web*. 11–18. https://doi.org/10.1145/775152.775155

[16] Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang, and Wei-Ying Ma. 2006. Simultaneous record detection and attribute labeling in web data extraction. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 494–503. https://doi.org/10.1145/1150402.1150457