# CoVA: Context-aware Visual Attention for Webpage Information Extraction

**Anurendra Kumar**[*]     **Keval Morabia**[*]     **Jingjin Wang**
**Kevin Chen-Chuan Chang**     **Alexander Schwing**

University of Illinois at Urbana-Champaign
`{ak32,morabia2,jingjin9,kcchang,aschwing}@illinois.edu`

## Abstract

Webpage information extraction (WIE) is an important step to create knowledge bases. For this, classical WIE methods leverage the Document Object Model (DOM) tree of a website. However, use of the DOM tree poses significant challenges as context and appearance are encoded in an abstract manner. To address this challenge we propose to reformulate WIE as a context-aware Webpage Object Detection task. Specifically, we develop a Context-aware Visual Attention-based (CoVA) detection pipeline which combines appearance features with syntactical structure from the DOM tree. To study the approach we collect a new large-scale dataset[1] of e-commerce websites for which we manually annotate every web element with four labels: product price, product title, product image and others. On this dataset we show that the proposed CoVA approach is a new challenging baseline which improves upon prior state-of-the-art methods.

## 1 Introduction

Webpage information extraction (WIE) is an important step when creating a large-scale knowledge base (Chang et al., 2006; Azir and Ahmad, 2017) which has many downstream applications such as knowledge-aware question answering (Lin et al., 2019) and recommendation systems (Ma et al., 2019; Lin et al., 2020).

Classical methods for WIE, like Wrapper Induction (Soderland, 1999; Muslea et al., 1998; Chang and Lui, 2001), rely on the publicly available source code of websites. The code is commonly parsed into a document object model (DOM) tree. The DOM tree is a programming language independent tree representation of any website, which contains all its elements. It can be obtained using various libraries like Puppeteer. These elements contain information about their location in the rendered webpage, styling like font size, etc., and text if it is a leaf node. State of the art method in WIE (Lin et al., 2020) uses text and markup information and employ CNN-BiLSTM encoder (Rhanoui et al., 2019) on the sequence of HTML nodes obtained from DOM to learn the embedding of each node.

However, using only the DOM tree for WIE is increasingly challenging for a variety of reasons: 1) Webpages are programmed to be aesthetically pleasing; 2) Oftentimes content and style is separated in website code and hence the DOM tree; 3) The same visual result can be obtained in a plethora of ways; 4) Branding banners and advertisements are interspersed with information of interest.

For this reason, recently, WIE applied optical character recognition (OCR) on rendered websites followed by word embedding-based natural language extraction (Staar et al., 2018). However, as mentioned before, recent webpages are highly enriched with visual content, and classical word embeddings don't capture this contextual information. For instance, text in advertising banners may be interpreted as valuable information. For this reason, a simple OCR detection followed by natural language processing techniques is a suboptimal for WIE (Vishwanath et al., 2018).

In response to these challenges we develop WIE based on a visual representation of a web element and its context. This permits to address the aforementioned four challenges. Moreover, visual features are independent of the programming language (*e.g.*, HTML for webpages, Dart for Android or iOS apps) and partially also the website language (*e.g.*, Arabic, Chinese, English). Intuitively, we aim to mimic the ability of humans to detect the location of target elements like product price, product title and product image on a webpage in a foreign language like the one shown in Fig. 1.

For this, we develop a context-aware Webpage

---

[1]CoVA dataset and code are available at github.com/kevalmorabia97/CoVA-Web-Object-Detection

[*]These authors contributed equally to this work

Figure 1: A person can detect the element for product price, title, and image, w/o knowing (a) Arabic or (b) Chinese

Object Detection (WOD), which we refer to as Context-aware Visual Attention-based detection (CoVA), where entities like prices are objects. Somewhat differently from an object in natural images which can be detected largely based on its appearance, objects on a webpage are strongly defined by contextual information. *e.g.*, a cat's appearance is largely independent of its nearby objects, whereas a product price is a highly ambiguous object (Fig. 2). It refers to the price of a product only when it is contextually related to a product title and a product image. The developed WOD uses a graph attention based architecture, which leverages the underlying syntactic DOM tree (Zhou et al., 2021) to focus on important context (Zhu et al., 2005) while classifying an element on a webpage. Once these web elements are identified, the relevant information *e.g.* price and title can be obtained from the corresponding DOM nodes. These information can then be indexed and used for applications like product search and price comparison across online retailers.

To facilitate this task we create a dataset of 7.7k English product webpage screenshots along with DOM information spanning 408 different websites (domains). We compare the results of CoVA with existing and newly created baselines that take visual features into account. We show that CoVA leads to substantial improvements while yielding interpretable contextual representations.

In summary, we make the following contributions:

1. We formulate WIE as a context-aware WOD problem.
2. We develop a Context-aware Visual Attention-based (CoVA) detection pipeline, which is end-to-end trainable and exploits syntactic

structure from the DOM tree along with screenshots. CoVA improves recent state-of-the-art baselines by a significant margin.
3. We create the largest public dataset of 7.7k English product webpage screenshots from 408 online retailers for Object Detection from product webpages. Our dataset is ∼ 10× larger than existing datasets.
4. We show the interpretability of CoVA using attention visualizations (Sec. 6.5)
5. We claim and validate that visual features (without textual content) along with DOM information are sufficient for many tasks while allowing cross-domain and cross-language generalizability. CoVA trained on English webpages perform well on Chinese Webpages (Sec. 6.4).

## 2 Related Work

**Webpage information extraction** (WIE) has been mainly addressed with Wrapper Induction (WI). WI aims to learn a set of extraction rules from HTML code or text, using manually labeled examples and counter-examples (Soderland, 1999; Muslea et al., 1998; Chang and Lui, 2001). These often require human intervention which is time-consuming, error-prone (Vadrevu et al., 2005), and does not generalize to new templates.

**Supervised learning**, which treats WIE as a classification task has also garnered significant attention. Traditionally, natural language processing techniques are employed over HTML or DOM information. Structural and semantic features (Ibrahim et al., 2008; Gibson et al., 2007) are obtained for each part of a webpage to predict categories like title, author, etc. Wu et al. (2015)
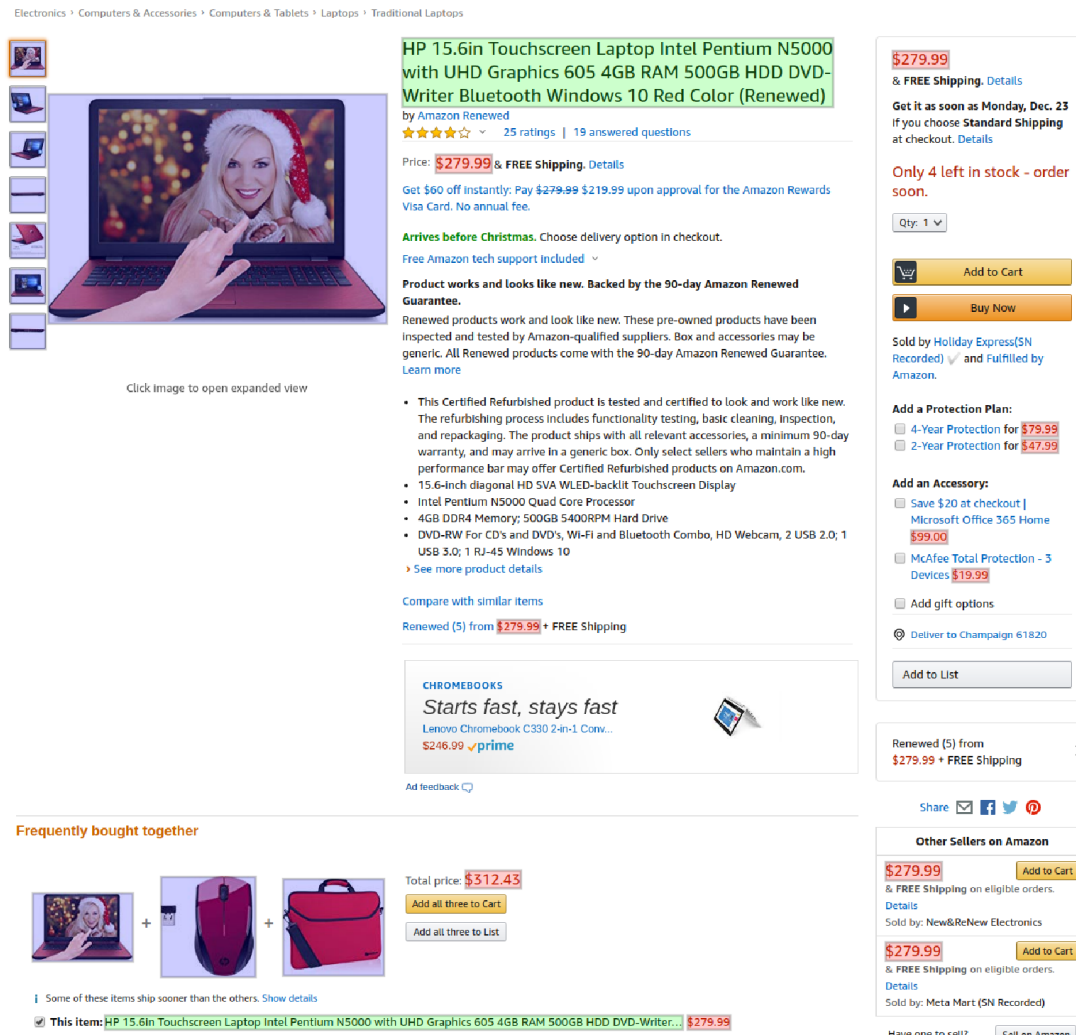
Figure 2: Example webpage showing multiple possible prices (red), but relatively fewer possible title (green) or image (purple)

casts WIE as a HTML node selection problem using features such as positions, areas, fonts, text, tags, and links. Lin et al. (2020) proposes a neural network to learn representation of a DOM node by combining text and markup information. A CNN-BiLSTM encoder is employed to learn the embeddings for HTML node. Hwang et al. (2020) develops a transformer architecture to learn spatial dependency between DOM nodes. Unlike these work which depends on text information, we aim to learn representation of a DOM node using only visual cues. Joshi and Liu (2009) develop a semantic similarity between blocks of webpages using textual and DOM features to extract the key article on a webpage.

**Visual features** have been extensively employed to generate visual wrappers for pattern extraction. Mostly, these utilize hand-crafted visual features from a webpage, *e.g.*, area size, font size, and type.

Cai et al. (2003) develop a visual block tree of a webpage using visual and layout features along with the DOM tree information. Subsequent works use this tree for tasks like webpage segmentation, visual wrapper generation, and web record extraction (Cai et al., 2004; Liu et al., 2003; Simon and Lausen, 2005; Burget and Rudolfova, 2009). Gogar et al. (2016) aims to develop domain-specific wrappers which generalize across unseen templates and don't need manual intervention. They develop a unified model that encodes visual, textual, and positional features using a single CNN.

**Object detection (OD) techniques in Computer Vision**, which aims to detect and classify all objects, has been extensively studied for natural images. Deep learning methods such as YOLO (Redmon and Farhadi, 2018), R-CNN variants (Girshick et al., 2014; Girshick, 2015; He et al., 2017), etc. yielded state-of-the-art results in OD.

**OD methods that can capture contextual information** are of particular interest here. Murphy et al. (2006) learn local and global context by object presence and localization and use a product of experts model (Hinton, 2002) to combine them. Kong et al. (2021) proposes a short path context module which transforms the integrated feature maps by considering local feature affinities.

**Graph Convolutional Networks (GCN)** (Kipf and Welling, 2016) was proposed to learn a node representation while taking neighbors of a node into account. Using it, Liu et al. (2019) represent a visually rich document as a complete graph of text content obtained by passing OCR (Mithe et al., 2013). They employ GCN to learn node representations for each web element.

Recently, **Attention mechanisms** have also shown remarkable ability in capturing contextual information (Bahdanau et al., 2014). Vaswani et al. (2017) propose a transformer architecture for language modeling. Luo et al. (2018) use attention over a BiLSTM-CRF layer for Named Entity Recognition (NER) on biomedical data. Word vectors learned on BERT (Devlin et al., 2018), which use self-attention, have yielded state-of-the-art results on 11 NLP tasks.

Separately, attention has been used for contextual learning in OD (Li et al., 2013; Hsieh et al., 2019; Morabia et al., 2020) and image captioning (You et al., 2016). Attention mechanisms have also been employed over graphs to learn an optimal representation of nodes while taking graph structure into account (Veličković et al., 2017). Moreover, attention permits to interpret result, which is often desired in many applications. We show our visualizations depicting this advantage below (Sec. 6.5).

## 3 Problem formulation

The DOM tree captures the syntactical structure of a webpage similar to a parse tree of a natural language. Our goal is to extract semantic information exploiting this syntactic structure. We view a leaf web element as a *word* and the webpage as a document with the DOM tree as its underlying parse tree. Formally, we represent a webpage $W$ as the set $W = \{v_1, v_2, \ldots, v_i, \ldots, v_N, D\}$ where $v_i$ denotes the visual representation of the $i$-th web element, $N$ denotes number of web elements, and $D$ refers to the DOM tree which contains the relations between the web elements. Our goal is to learn a parametric function $f_\theta(y_i|W, i)$

which extracts a visual representation $v_i$ of the $i$-th web element from website $W$ so as to accurately predict label $y_i$ of the web element. In the following we consider four labels for a product, *i.e.*, $y_i \in \{\text{product price}, \text{title}, \text{image}, \text{others}\}$. The parameters $\theta$ are obtained by minimizing the following supervised classification loss

$$\theta^* = \underset{\theta}{\arg\min} \quad \underset{i, W \sim P_W}{\mathbb{E}} \left[ \mathcal{L}(f_\theta(y_i|W, i), y_i^*) \right],$$

where $\mathbb{E}$ denotes an expectation, $y_i$ and $y_i^*$ denote the predicted and ground truth labels and $P_W$ denotes a probability distribution over webpages.

Information of a webpage is present in the leaves of the DOM tree, *i.e.*, the web elements $i$. Web elements are an atomic entity which is characterized by a rectangular bounding box. We can extract the target information $y_i$ from the DOM tree if we know the exact leaf bounding boxes of the desired element. Therefore, we can view WIE as an object detection (OD) task where objects are leaf elements and might contain the desired entity (target). However, identity $y_i$ of a web element is heavily dependent on its context, *e.g.*, price, title, and image of a product are most likely to be in same or nearby sub-tree in comparison to unrelated web elements such as advertisements. Similarly, there can be multiple instances of price-like elements. However, the correct price would be contextually positioned with product title and image (Fig. 2). Therefore, we formulate WIE as a context-aware OD.

We use the DOM tree to identify context for a web element. We represent the syntactic closeness between web elements through edges in the graph (discussed in next section). We then employ a graph attention mechanism (Veličković et al., 2017) to attend to the most important contexts.

## 4 Proposed End-to-End Pipeline – CoVA

In this section, we present our Context-Aware Visual Attention-based end-to-end pipeline for Webpage Object Detection (CoVA) which aims to learn function $f$ to predict labels $y = [y_1, y_2, \ldots, y_N]$ for a webpage. The input to CoVA consists of 1. a screenshot of a webpage, 2. list of bounding boxes $[x, y, w, h]$ of the web elements, and 3. neighborhood information for each element obtained from DOM. It should be noted that bounding boxes of the web elements are relatively accurate and doesn't pose challenges similar to OD for natural images.
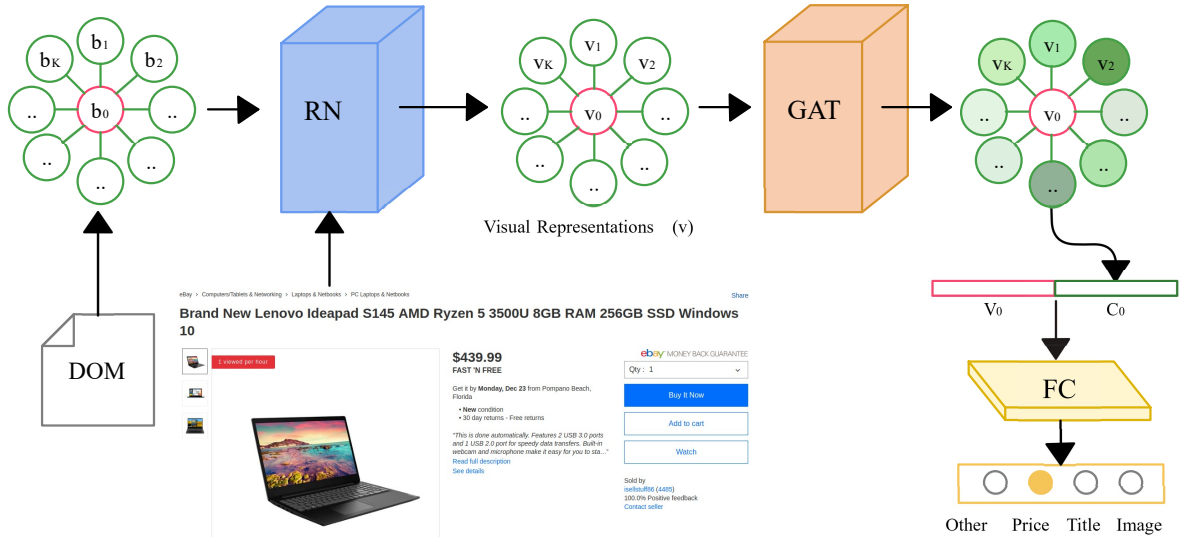
Figure 3: CoVA end-to-end training pipeline (for a single web element). CoVA takes a webpage screenshot and list of bounding boxes along with $K$ neighbors for each web element (obtained from DOM). RN learns visual representation ($v_0$) while GAT learns contextual representation ($c_0$) from its neighbor's visual representations.

As illustrated in Fig. 3, this information is processed by CoVA in four stages: 1. the graph representation extraction for the webpage, 2. the Representation Network (RN), 3. the Graph Attention Network (GAT), and 4. a fully connected (FC) layer. The graph representation extraction computes for every web element $i$ its set of neighboring web elements $\mathcal{N}_i$. The RN consists of a Convolutional Neural Net (CNN) and a positional encoder aimed to learn a visual representation $v_i$ for each web element $i \in \{1, \ldots, N\}$. The GAT combines the visual representation $v_i$ of the web element $i$ to be classified and those of its neighbors, *i.e.*, $v_k$ $\forall k \in \mathcal{N}_i$ to compute the contextual representation $c_i$ for web element $i$. Finally, the visual and contextual representations of the web element are concatenated and passed through the FC layer to obtain the classification output. We describe each of the components next.

## 4.1 Webpage as a Graph

We represent a webpage as a graph where nodes are leaf web elements and an edge indicates that the corresponding web elements are contextually relevant to each other. A naive way to create graph is by putting edge between every pair of nodes (Liu et al., 2019). An alternative way of creating a graph is to add edges to nearby nodes based on spatial distance. However, web elements vary greatly in shapes & sizes, and two web elements might have small distance but they're contextually irrelevant since they lie in different DOM subtrees. For this, we use the $K$ nearest leaf elements in the DOM

tree as the neighbors $\mathcal{N}_i$ a web element $i$. An edge within the graph denotes the syntactic closeness in the DOM tree.

## 4.2 Representation Network (RN)

The goal of the Representation Network (RN) is to learn a fixed size visual representation $v_i$ of any web element $i \in \{1, \ldots, N\}$. This is important since web elements have different sizes, aspect ratios, and content type (image or text). To achieve this the RN consists of a CNN operating on the screenshot of a webpage, followed by a Region of Interest (RoI) pooling layer (Girshick, 2015) and a positional encoder. Specifically, RoI pooling is performed to obtain a fixed size representation for all web elements. To capture the spatial layout, we learn a $P$ dimensional positional feature which is obtained by passing the bounding box features $[x, y, w, h, \frac{w}{h}]$ through a positional encoder implemented by a single layer neural net. Finally, we concatenate the flattened output of the RoI pooling with positional features to obtain the visual representation $v_i$.

## 4.3 Graph Attention Network (GAT)

The goal of the graph attention network is to compute a contextual representation $c_i$ for each web element $i$ which takes visual information $v_i$ from neighboring web elements into account. However, out of multiple neighbors for a web element, only a few are informative, *e.g.*, a web element having a currency symbol near a set of digits seems relevant. To identify the relational importance we

use a Graph Attention Network (GAT) (Veličković et al., 2017). We transform each of the input features by learning projection matrices $W_1$ and $W_2$ applied at every node and its neighbors. We then employ self-attention (Lin et al., 2017) to compute the importance score,

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(a^T[W_1v_i||W_2v_j]))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(a^T[W_1v_i||W_2v_k]))},$$

where $\cdot^T$ represents transposition, $||$ is the concatenation operation, $\mathcal{N}_i$ denotes the neighbors of web element $i$. The weights $\alpha_{ij}$ are non-negative attention scores for neighboring web elements of web element $i$. Finally, we obtain the contextual representation $c_i$ for a web element $i$ as a weighted combination of projected visual representations of its neighbors, *i.e.*, via

$$c_i = \sum_{j \in \mathcal{N}_i} \alpha_{ij} W_2 v_j. \tag{1}$$

### 4.4 Augmenting CoVA with extra features

In scenarios where additional features (*e.g.*, text content, HTML tag information, etc.) are available, CoVA can be easily extended to incorporate those. These features can be concatenated with visual representations obtained from the RN without modifying the pipeline in any other way. We refer to this extended pipeline as **CoVA++**. However, making the model dependent on these features might lead to constraints regarding the programming language (HTML tags) or text language. In Sec. 6.4, we show that CoVA trained on English webpages (without additional features) generalizes well to Chinese webpages.

## 5 Dataset Generation

To the best of our knowledge there is no large-scale dataset for WIE with visual annotations for object detection. So far, the Structured Web Data Extraction (SWDE) dataset (Hao et al., 2011) is the only known large dataset that can be used for training deep neural networks for WIE (Lin et al., 2020; Lockard et al., 2019). SWDE dataset contains webpage HTML codes which is not sufficient to render it into a screenshot (since it contains links to old and non-existent URLs). Because of this we create a new large-scale labeled dataset for object detection on English product webpage screenshots along with DOM information. We chose e-commerce websites since those have been a de-facto standard

for WIE (Gogar et al., 2016; Zhu et al., 2005). Our dataset generation consists of two steps: 1. search the web with 'shopping' keywords to aggregate diverse webpages and employ heuristics to automate labeling of product price, title, and image, 2. manual correction of incorrect labels. We discuss both steps next.

**Web scraping and coarse labeling.** To scrape websites, we use Google shopping[2] which aggregates links to multiple online retailers (domains) for the same product. These links are uploaded by the merchants of the respective domains. We do a keyword search for various categories, like electronics, food, cosmetics. For each search result, we record the price and title from Google shopping. Then, we navigate through the links to specific product websites and save a $1280 \times 1280$ screenshot. To extract a bounding box for each web element, we store a pruned DOM tree. Price and title candidates are labeled by comparing with the recorded values using heuristics. For product images, we always choose the DOM element having the largest bounding box area among all the elements with an <img> HTML tag, although this might not be true for many websites. We correct this issue in the next step.

**Label correction.** The coarse labeling is only $\sim 60\%$ accurate because 1. price on webpages keeps changing and might differ from the Google shopping price, and 2. many bounding boxes have the same content. To correct for these mistakes, we manually inspected and correct labeling errors. We obtained **7,740 webpages** spanning **408 domains**. Each of these webpages contains exactly one labeled price, title, and image. All other web elements are labeled as 'others'. On average, there are $\sim 90$ leaf web elements on a webpage.

**Train-Val-Test split.** We create a cross-domain split which ensures that each of the train, val and test sets contains webpages from different domains. We observed that the top-5 frequent domains were Amazon, EBay, Walmart, Etsy, and Target. So, we created 5 different splits for 5-Fold Cross Validation such that each of the major domains is present in one of the test splits.

## 6 Experimental Setup & Results

### 6.1 Baseline Methods

We compare the results of our end-to-end pipeline CoVA with other existing and newly created base-
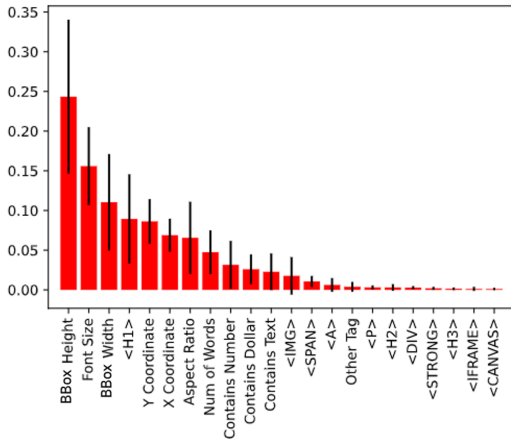
---
[2]shopping.google.com

Figure 4: Gini impurity-based importance of features in RF

lines summarized below. Our newly created baselines combine existing object detection and graph based models to identify the importance of visual features and contextual representations.

**(Gogar et al., 2016):** This method identifies product price, title, and image from the visual and textual representation of the web elements.

**Random Forest on Heuristic features:** We train a Random Forest classifier with 100 trees using various HTML tags, text, and bounding box features as shown in Fig. 4.

**Fast R-CNN\*:** We compare with Fast R-CNN (Girshick, 2015) to quantify the importance of contextual representations in CoVA. We use the DOM tree instead of selective search (Uijlings et al., 2013) for bounding box proposals. We also use positional features as described when discussing the representation network (Sec. 4.2) for a fair comparison with CoVA. We will refer to this baseline as 'Fast R-CNN\*.'

**Fast R-CNN\* + GCN (Kipf and Welling, 2016):** We use GCN on our graph formulation where node features are the visual representations obtained from Fast R-CNN\*.

**Fast R-CNN\* + Bi-LSTM (Schuster and Paliwal, 1997):** We train a bidirectional LSTM on visual representations of web elements in preorder traversal of the DOM tree. We use its output as the contextual representation and concatenate it with the visual representation of the web element obtained from Fast R-CNN\*.

### 6.2 Model Training, Inference and Evaluation

In each training epoch, we randomly sample $90\%$ from others. This increases the diversity in training data by providing different contexts for web-

pages with exactly the same template. We use batch normalization (Ioffe and Szegedy, 2015) between consecutive layers, Adam optimizer for updating model parameters and minimize cross-entropy loss. During inference, the model detects one web element with highest probability for each class. Once the web element is identified, the corresponding text content can be extracted from the DOM tree or by using OCR for downstream tasks.

For CoVA++ we use as additional information the same heuristic features used to train the Random Forest classifier baseline. Unless specified otherwise, all results of CoVA and baselines use the following hyperparameters where applicable: learning rate = 5e-4, batch size = 5 screenshot images, $K = 24$ neighbor elements in the graph, RoI pool output size $(H \times W) = (3 \times 3)$, dropout = 0.2, $P = 32$ dimensional positional features, output dimension for projection matrix $W_1, W_2$ is 384, weight decay = 1e-3. We use the first 5 layers of a pre-trained ResNet18 (He et al., 2016) in the representation network (RN), which yields a 64 channel feature map. This significantly reduces the parameters in the RN from $12m$ to $0.2m$ and speeds up training at the same time. The evaluation is performed using Cross-domain Accuracy for each class, *i.e.*, the fraction of webpages of new domains with correct class. All the experiments are performed on Tesla V100-SXM2-16GB GPUs.

### 6.3 Results

As shown in Table 1, our method outperforms all baselines by a considerable margin especially for price prediction. CoVA learns visual features which are significantly better than the heuristic feature baseline that uses predefined tag, textual and visual features. Fig. 4 shows the importance of different heuristic based features in a webpage. We observe that a heuristic feature based method has similar performance to methods which don't use contextual features. Moreover, CoVA++ which also uses heuristic features, doesn't lead to statistically significant improvements. This shows that visual features learnt by CoVA are more general for tasks like price & title detection. Context information is particularly important for price (in comparison to title and image) since it's highly ambiguous and occurs in different locations with varying contexts (Fig. 2). This is evident from the $\sim 8.9\%$ improvement in price accuracy compared to the Fast R-CNN\*. Unless stated other-

| Method | Params | Price Acc | Title Acc | Image Acc |
|---|---|---|---|---|
| Gogar et al. (2016) | 1.8m | $78.1 \pm 17.2$ | $91.5 \pm 1.3$ | $93.2 \pm 1.9$ |
| Random Forest using Heuristic features | - | $87.4 \pm 10.4$ | $93.5 \pm 5.3$ | $97.2 \pm 3.8$ |
| Fast R-CNN* (Girshick, 2015) | 0.5m | $86.6 \pm 7.3$ | $93.7 \pm 2.2$ | $97.0 \pm 3.6$ |
| Fast R-CNN* + GCN | 1.4m | $90.0 \pm 11.0$ | $95.4 \pm 1.5$ | $98.2 \pm 2.8$ |
| Fast R-CNN* + Bi-LSTM | 5.1m | $92.9 \pm 4.6$ | $94.0 \pm 2.1$ | $97.6 \pm 3.6$ |
| **CoVA** | 1.6m | $\mathbf{95.5 \pm 3.8}$ | $\mathbf{95.7 \pm 1.2}$ | $\mathbf{98.8 \pm 1.5}$ |
| **CoVA++** | 1.7m | $\mathbf{96.1 \pm 3.0}$ | $\mathbf{96.7 \pm 2.2}$ | $\mathbf{99.6 \pm 0.3}$ |

Table 1: Cross Domain Accuracy (mean $\pm$ standard deviation) for 5-fold cross validation.

wise, we will discuss results with respect to price accuracy. We observe that CoVA yields stable results across folds ($\sim 3.5\%$ reduction in standard deviation). This shows that CoVA learns features which are generalizable and which have less dependence on the training data. Using GCN with Fast R-CNN* leads to unstable results with $11\%$ standard deviation while yielding a $3.4\%$ improvement over Fast R-CNN*. Fast R-CNN* with Bi-LSTM is able to summarize the contextual features by yielding a $\sim 6.3\%$ improvement in comparison to Fast RCNN*. CoVA outperforms Fast RCNN* with Bi-LSTM by $\sim 2.6\%$ with much fewer number of parameters while also yielding interpretable results. We also obtained top-3 accuracy for CoVA, which are $98.6\%$, $99.4\%$, and $99.9\%$ for price, title and image respectively.

### 6.4 Cross-lingual Evaluation of CoVA

To validate our claim that visual features (without textual or HTML tag information) can capture cross-lingual information, we test our model on webpages in a foreign language. In particular, we evaluated CoVA (trained on English product webpages) using 100 Chinese product webpages spanning across 25 unique domains. CoVA achieves $92\%$, $90\%$, and $99\%$ accuracy for product price, title, and image. It should be noted that image has the same accuracy as for English pages. This is expected since images have no language components that the model can attend to.

### 6.5 Attention Visualizations

Table 1 shows that attention significantly improves performance for all the three targets. As discussed earlier, only few of the contexts are important which are effectively learnt by Graph Attention Network (GAT). We observed that on average, $\sim 20\%$ of context elements were activated (score above 0.05 threshold) by GAT. We also study a multihead

attention instead of single head following (Vaswani et al., 2017), which didn't yield significant improvements in our case.

Fig. 5 shows visualizations of attention scores learnt by GAT. Fig. 5(a) shows an example where title and image have more weight than other contexts when learning a context representation for price. This shows that attention is able to focus on important web elements and discards others. Similarly, Fig. 5(b) shows that price has a much higher score than other contexts for learning contextual representation for title.

## 7 Ablation Studies

**Importance of Positional features:** Table 2 shows that positional features can significantly improve accuracy for price, title, and image prediction. This also validates that for webpage OD, location and size of a bounding box carries significant information, making it different from classical OD.

**Dependence on number of neighbors in graph:** Fig. 6 shows the variation in cross domain accuracy of CoVA with respect to the number of neighboring elements $K$. Note that having 0 context elements is equivalent to our baseline Fast R-CNN*. We observe that, unlike title and image, price accuracy can significantly be improved by considering larger contexts. This is due to the fact that price is highly ambiguous (Fig. 2). We also study the graph construction described by (Liu et al., 2019) where all nodes are considered in the neighborhood of a particular node. This significantly reduced the performance for price ($90.7\%$) and title ($92.7\%$).

## 8 Conclusion & Future Work

In this paper, we reformulated the problem of webpage IE (WIE) as a context-aware webpage object detection. We created a large-scale dataset for this task and is available publicly. We proposed CoVA
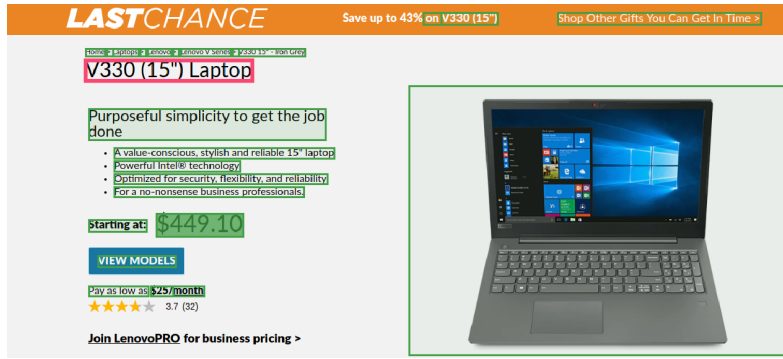
(a)
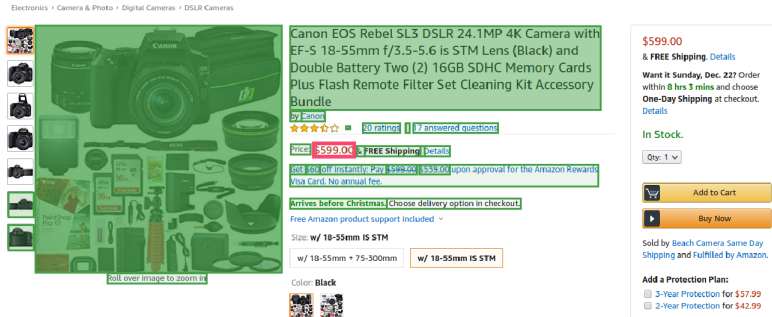


(b)

Figure 5: Attention Visualizations where red border denotes web element to be classified, and its contexts have green shade whose intensity denotes score. Price in (a) get much more score than other contexts. Title and image in (b) are scored higher than other contexts for price.

| Method | Price Accuracy | Title Accuracy | Image Accuracy |
|---|---|---|---|
| CoVA without positional features | $89.2 \pm 10.3$ | $91.9 \pm 1.4$ | $95.9 \pm 1.8$ |
| **CoVA** | $\mathbf{95.5 \pm 3.8}$ | $\mathbf{95.7 \pm 1.2}$ | $\mathbf{98.8 \pm 1.5}$ |

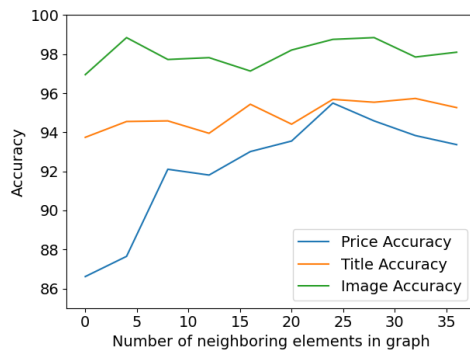Table 2: Importance of positional features in RN



Figure 6: Comparison of context size with accuracy

which uses i) a graph representation of a webpage, ii) a Representation Network (RN) to learn visual representation for a web element, and iii) a Graph Attention Network (GAT) for contextual learning. CoVA improves upon state-of-the-art results and newly created baselines by considerable margins. Our visualizations show that CoVA is able to attend to the most important contexts. In the future, we

plan to adapt this method to other tasks such as identifying malicious web elements. Our works shows the importance of visual features of WIE which is traditionally overlooked. We hope that our work will motivate researchers in WIE to employ CV alongwith NLP techniques to solve this important problem.

## References

Mohd Amir Bin Mohd Azir and Kamsuriah Binti Ahmad. 2017. Wrapper approaches for web data extraction: A review. In *2017 6th International Conference on Electrical Engineering and Informatics (ICEEI)*, pages 1–6. IEEE.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Radek Burget and Ivana Rudolfova. 2009. Web page element classification based on visual features. In

*2009 First Asian Conference on Intelligent Information and Database Systems*, pages 67–72. IEEE.

Deng Cai, Xiaofei He, Ji-Rong Wen, and Wei-Ying Ma. 2004. Block-level link analysis. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 440–447.

Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. 2003. Vips: a vision-based page segmentation algorithm.

Chia-Hui Chang, Mohammed Kayed, Moheb R Girgis, and Khaled F Shaalan. 2006. A survey of web information extraction systems. *IEEE transactions on knowledge and data engineering*, 18(10):1411–1428.

Chia-Hui Chang and Shao-Chen Lui. 2001. Iepad: information extraction based on pattern discovery. In *Proceedings of the 10th international conference on World Wide Web*, pages 681–688.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

John Gibson, Ben Wellner, and Susan Lubar. 2007. Adaptive web-page content identification. In *Proceedings of the 9th annual ACM international workshop on Web information and data management*, pages 105–112.

Ross Girshick. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

Tomas Gogar, Ondrej Hubacek, and Jan Sedivy. 2016. Deep neural networks for web page information extraction. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 154–163. Springer.

Qiang Hao, Rui Cai, Yanwei Pang, and Lei Zhang. 2011. From one tree to a forest: a unified solution for structured web data extraction. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 775–784.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. 2019. One-shot object detection with co-attention and co-excitation. In *Advances in Neural Information Processing Systems*, pages 2725–2734.

Wonseok Hwang, Jinyeong Yim, Seunghyun Park, Sohee Yang, and Minjoon Seo. 2020. Spatial dependency parsing for semi-structured document information extraction. *arXiv preprint arXiv:2005.00642*.

Hossam Ibrahim, Kareem Darwish, and Abdel-Rahim Madany. 2008. Automatic extraction of textual elements from news web pages. In *LREC*.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Parag Mulendra Joshi and Sam Liu. 2009. Web document text and images extraction using dom analysis and natural language processing. In *Proceedings of the 9th ACM symposium on Document engineering*, pages 218–221.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Yuqiu Kong, Mengyang Feng, Xin Li, Huchuan Lu, Xiuping Liu, and Baocai Yin. 2021. Spatial context-aware network for salient object detection. *Pattern Recognition*, 114:107867.

Gui Li, Cheng Chen, Zheng Yu Li, Zi Yang Han, and Ping Sun. 2013. Web data extraction based on tag path clustering. In *Advanced Materials Research*, volume 756, pages 1590–1594. Trans Tech Publ.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151*.

Bill Yuchen Lin, Ying Sheng, Nguyen Vo, and Sandeep Tata. 2020. Freedom: A transferable neural architecture for structured information extraction on web documents. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1092–1102.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Bing Liu, Robert Grossman, and Yanhong Zhai. 2003. Mining data records in web pages. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–606.

Xiaojing Liu, Feiyu Gao, Qiong Zhang, and Huasha Zhao. 2019. Graph convolution for multimodal information extraction from visually rich documents. *arXiv preprint arXiv:1903.11279*.

Colin Lockard, Prashant Shiralkar, and Xin Luna Dong. 2019. OpenCeres: When open information extraction meets the semi-structured web. 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)*, pages 3047–3056, Minneapolis, Minnesota. Association for Computational Linguistics.

Ling Luo, Zhihao Yang, Pei Yang, Yin Zhang, Lei Wang, Hongfei Lin, and Jian Wang. 2018. An attention-based bilstm-crf approach to document-level chemical named entity recognition. *Bioinformatics*, 34(8):1381–1388.

Weizhi Ma, Min Zhang, Yue Cao, Woojeong Jin, Chenyang Wang, Yiqun Liu, Shaoping Ma, and Xiang Ren. 2019. Jointly learning explainable rules for recommendation with knowledge graph. In *The World Wide Web Conference*, pages 1210–1221.

Ravina Mithe, Supriya Indalkar, and Nilam Divekar. 2013. Optical character recognition. *International journal of recent technology and engineering (IJRTE)*, 2(1):72–75.

Keval Morabia, Jatin Arora, and Tara Vijaykumar. 2020. Attention-based joint detection of object and semantic part. *arXiv preprint arXiv:2007.02419*.

Kevin Murphy, Antonio Torralba, Daniel Eaton, and William Freeman. 2006. Object detection and localization using local and global features. In *Toward Category-Level Object Recognition*, pages 382–400. Springer.

Ion Muslea, Steve Minton, and Craig Knoblock. 1998. Stalker: Learning extraction rules for semistructured, web-based information sources. In *Proceedings of AAAI-98 Workshop on AI and Information Integration*, pages 74–81. AAAI Press.

Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Maryem Rhanoui, Mounia Mikram, Siham Yousfi, and Soukaina Barzali. 2019. A cnn-bilstm model for document-level sentiment analysis. *Machine Learning and Knowledge Extraction*, 1(3):832–847.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

Kai Simon and Georg Lausen. 2005. Viper: augmenting automatic information extraction with visual perceptions. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 381–388.

Stephen Soderland. 1999. Learning information extraction rules for semi-structured and free text. *Machine learning*, 34(1-3):233–272.

Peter WJ Staar, Michele Dolfi, Christoph Auer, and Costas Bekas. 2018. Corpus conversion service: A machine learning platform to ingest documents at scale. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 774–782.

Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. 2013. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171.

Srinivas Vadrevu, Saravanakumar Nagarajan, Fatih Gelgi, and Hasan Davulcu. 2005. Automated metadata and instance extraction from news web sites. In *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, pages 38–41. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

D Vishwanath, Rohit Rahul, Gunjan Sehgal, Arindam Chowdhury, Monika Sharma, Lovekesh Vig, Gautam Shroff, Ashwin Srinivasan, et al. 2018. Deep reader: Information extraction from document images via relation extraction and natural language. In *Asian Conference on Computer Vision*, pages 186–201. Springer.

Shanchan Wu, Jerry Liu, and Jian Fan. 2015. Automatic web content extraction by combination of learning and grouping. In *Proceedings of the 24th international conference on World Wide Web*, pages 1264–1274.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659.

Yichao Zhou, Ying Sheng, Nguyen Vo, Nick Edmonds, and Sandeep Tata. 2021. Simplified dom trees for transferable attribute extraction from the web. *arXiv preprint arXiv:2101.02415*.

Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang, and Wei-Ying Ma. 2005. 2d conditional random fields for web information extraction. In *Proceedings of the 22nd international conference on Machine learning*, pages 1044–1051.