Form2Seq : A Framework for Higher-Order Form Structure Extraction

Milan Aggarwal[1], **Hiresh Gupta**[2], **Mausoom Sarkar**[1], **Balaji Krishnamurthy**[1]
Media and Data Science Research Labs, Adobe[1]
Adobe Experience Cloud[2]

Abstract

Document structure extraction has been a widely researched area for decades with recent works performing it as a semantic segmentation task over document images using fully-convolution networks. Such methods are limited by image resolution due to which they fail to disambiguate structures in dense regions which appear commonly in forms. To mitigate this, we propose Form2Seq, a novel sequence-to-sequence (Seq2Seq) inspired framework for structure extraction using text, with a specific focus on forms, which leverages relative spatial arrangement of structures. We discuss two tasks; 1) Classification of low-level constituent elements (TextBlock and empty fillable Widget) into ten types such as field captions, list items, and others; 2) Grouping lower-level elements into higher-order constructs, such as Text Fields, ChoiceFields and ChoiceGroups, used as information collection mechanism in forms. To achieve this, we arrange the constituent elements linearly in natural reading order, feed their spatial and textual representations to Seq2Seq framework, which sequentially outputs prediction of each element depending on the final task. We modify Seq2Seq for grouping task and discuss improvements obtained through cascaded end-to-end training of two tasks versus training in isolation. Experimental results show the effectiveness of our text-based approach achieving an accuracy of 90% on classification task and an F1 of 75.82, 86.01, 61.63 on groups discussed above respectively, outperforming segmentation baselines. Further we show our framework achieves state of the results for table structure recognition on ICDAR 2013 dataset.

1 Introduction

Various works (Hao et al., 2016; He et al., 2017; Wick and Puppe, 2018; Yang et al., 2017) have studied semantic structure extraction for documents. Structure extraction is necessary for digitizing documents to make them re-flowable and index-able, which is useful in web-based services (Alam and Rahman, 2003; Gupta et al., 2007; Khemakhem et al., 2018; Rahman and Alam, 2003). In this work, we look at a complex class of documents i.e., Forms that are used to capture user data by organizations across various domains such as government services, finance, administration, and healthcare. Such industries that have been using paper or PDF forms would want to convert them into an appropriate digitized version (Rahman and Alam, 2003) (such as an HTML). Once these forms are made re-flowable, they can be made available across devices with different form factors(Alam and Rahman, 2003; Gupta et al., 2007). This facilitates providing better form filling experiences and increases the ease of doing business since their users can interact with forms more conveniently and enables other capabilities like improved handling of filled data, applying validation checks on data filled in fields, consistent form design control[1].

To enable **dynamic rendering** of a form while **re-flowing** it, we need to extract its structure at multiple levels of hierarchy. We define TextBlock to be a logical block of self contained text. Widgets are spaces provided to fill information. Some low level elementary structures such as text and widgets can be extracted using auto-tagging capabilities of tools like Acrobat from the form PDF. However, such PDFs do not contain data about higher-order structures such as Text Fields, ChoiceGroups etc.

Document structure extraction has been studied extensively with recent works employing deep learning based fully convolution neural networks (He et al., 2017; Wick and Puppe, 2018; Yang et al., 2017) that perform semantic segmentation (Long et al., 2015; Chen et al., 2014; Noh et al., 2015) on

[1]Please refer to supplementary for re-flow visualisation

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 3830–3840,
November 16–20, 2020.

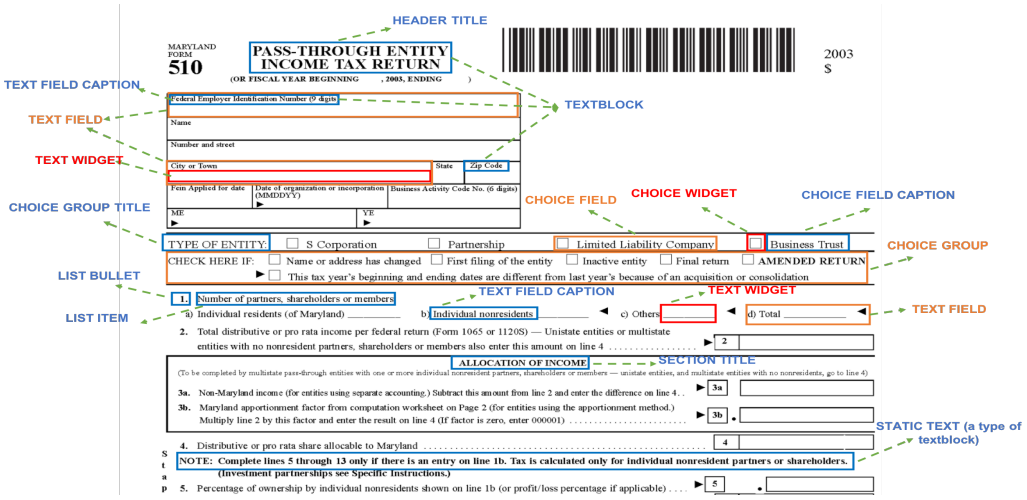

Figure 1: Different types of TextBlocks(Blue), Widgets(Red) & higher order groups(orange) - ChoiceGroups, Choice Fields, Text Fields in a form. A text field comprises of 1) textblock(referred as text field caption) that describes what to fill & 2) collection of widgets(text widgets). A choice group comprises of a title & collection of choice fields. Textblocks & widgets are classified into different types based on higher order group they are part of.

document image. Such methods perform well at extracting coarser structures but fail to extract closely spaced structures in form images (as discussed in the Experiments section). With increase in image resolution, number of activations(forward pass) and gradients(backward pass) increase at each network layer which requires more GPU memory during training. Since GPU memory is limited, they downscale the original image at the input layer which makes it difficult to disambiguate closely spaced structures, especially in dense regions(occurring commonly in forms) which leads to merging.

Figure 1 shows different types of TextBlocks, Widgets and higher order groups. Given text blocks and widgets as input, our Form2Seq framework classifies them between different type categories. We hypothesize that type classification of lower level elements can provide useful cues for extracting higher order constructs which are comprised of such smaller elements. We establish our hypothesis for the task of extracting ChoiceGroups, Text Fields and Choice Fields. A Text Field is composed of textblock(textual caption) and associated widgets, as shown in figure 1. A choice group is a collection of boolean fields called choice fields with an optional title text (choice group title) that describes instructions regarding filling it. We study fillable constructs as they are intrinsic and unique to forms and contain diverse elementary structures.

The spatial arrangement of lower level elements with respect to other elements in a form are correlated according to the type of construct. For instance, a list item usually follows a bullet in the reading order; field widgets are located near the field caption. Similarly, elements that are part of same higher-order group tend to be arranged in a spatially co-located manner. To leverage this in our Form2Seq framework, we perform a bottom up approach where we first classify lower level elements into different types. We arrange these elements in natural reading order to obtain a linear sequence. This sequence is fed to Seq2Seq (Sutskever et al., 2014) where each element's text and spatial representation is passed through a BiLSTM. The output of BiLSTM for each element is sequentially given as input to an LSTM (Hochreiter and Schmidhuber, 1997) based decoder which is trained to predict the category type. For grouping task, we modify the framework to predict id of the group each lower level element is part of. Here the model is trained to predict same group id for elements that are part of same group. Our contributions can be listed as:

- We propose Form2Seq framework for forms structure extraction, specifically for the tasks of element type classification and higher order group extraction.
- We show effectiveness of end-to-end training of both tasks through our proposed framework over performing group extraction alone.
- We perform ablations to establish role of text in improving performance on both tasks. Our approach outperforms image segmentation baselines.

- Further, we perform table structure recognition by grouping table text into rows and columns achieving state of the art results on ICDAR 2013 dataset.

2 Related Work

Earlier works for document layout analysis have mostly been rule based relying on hand crafted features for extracting coarser structures such as graphics and text paragraphs (Lebourgeois et al., 1992). Approaches like connected components and others, were also used for extracting text areas(Ha et al., 1995a) and physical layouts(Simon et al., 1997). These approaches can be classified into top-down (Ha et al., 1995b) or bottom-up (Drivas and Amin, 1995). The bottom-up methods focus on extracting text-lines and aggregating them into paragraphs. Top-down approaches detect layout by subdividing the page into blocks and columns.

With the advancement in deep learning, recent approaches have mostly been fully convolution neural network (FCN) based that eliminate need of designing complex heuristics (Yang et al., 2017; He et al., 2017; Wick and Puppe, 2018). FCNs were successfully trained for semantic segmentation (Long et al., 2015) which has now become a common technique for page segmentation. The high level feature representations make FCN effective for pixel-wise prediction. FCN has been used to locate/recognize handwritten annotations, particularly in historical documents (Kölsch et al., 2018). Wigington et al. proposed a model that jointly learns handwritten text detection and recognition using a region proposal network that detects text start positions and a line follow module which incrementally predicts the text line that should be subsequently used for reading.

Several methods have addressed regions in documents other than text such as tables, figures etc. Initial deep learning work that achieved success in table detection relied on selecting table like regions on basis of loose rules which are subsequently filtered by a CNN (Hao et al., 2016). He et al. proposed multi-scale, multi-task FCN comprising of two branches to detect contours in addition to page segmentation output that included tables. They additionally use CRF (Conditional Random Field) to make the segmented output smoother. However, segmentation based methods fail to disambiguate closely spaced structures in form images due to resolution limitations as discussed in experiments section. Graliński et al. introduced the new task of recognising only useful entities in long documents on two new datasets. FUNSD (Jaume et al., 2019) is a small-scale dataset for form understanding comprising of 200 annotated forms. In comparison, our Forms Dataset is much larger having richer set of annotations. For task of figure extraction from scientific documents, (Siegel et al., 2018) introduced a large scale dataset comprising of 5.5 million document labels. They find bounding boxes for figures in PDF by training Overfeat (Sermanet et al., 2013) on image embeddings generated using ResNet-101.

Few works have explored alternate input modalities such as text for other document related tasks. Extracting pre-defined and commonly occurring named entities from invoices like documents(using text and box coordinates) has been the main focus for some prior works (Katti et al., 2018; Liu et al., 2019; Denk and Reisswig, 2019; Majumder et al., 2020). Text and document layouts have been used for learning BERT (Devlin et al., 2019) like representations through pre-training and then combined with image features for information extraction from documents (Xu et al., 2020; Garncarek et al., 2020). However, our work focuses on extracting a much more generic, diverse, complex, dense, and hierarchical document structure from Forms. Document classification is a partly related problem that has been studied using CNN-only approaches for document verification (Sicre et al., 2017). Yang et al. have designed HAN which hierarchically builds sentence embeddings and then document representation using multi-level attention mechanism. Other works explored multi-modal approaches, using MobileNet (Howard et al., 2017) and FastText (Bojanowski et al., 2017) to extract visual and text features respectively, which are combined in different ways (such as concatenation) for document classification (Audebert et al., 2020). In contrast, we tackle a different task of form layout extraction which requires recognising different structures.

Yang et al. also proposed a multimodal FCN (MFCN) to segment figures, tables, lists etc. in addition to paragraphs from documents. They concatenate a text embedding map to feature volume. We consider image based semantic segmentation approaches as baselines for the tasks proposed. We compare the performance of our approach with 1) their FCN based method and 2) DeepLabV3+ (Chen et al., 2018), which is state of the art deep learning model for semantic segmentation.

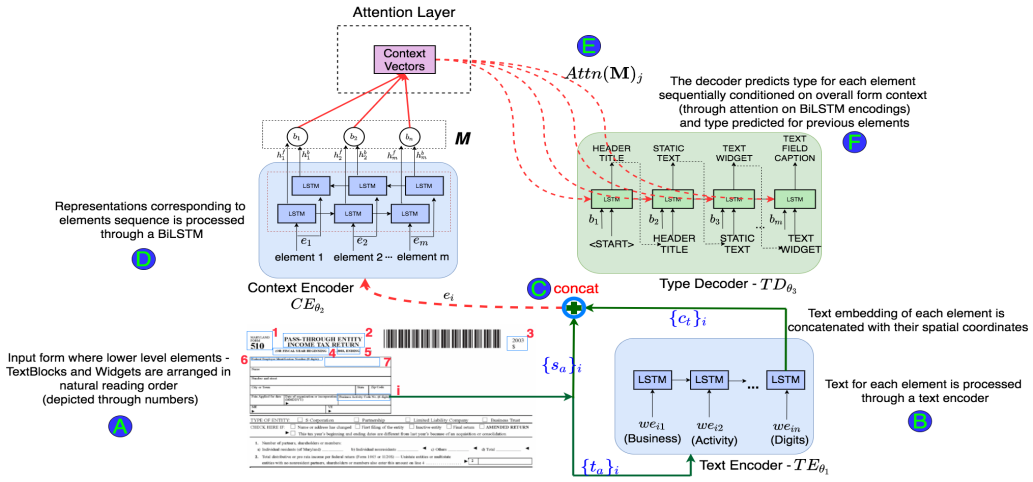

Figure 2: Model Architecture for element type classification. Different stages are annotated with letters.

3 Methodology

The spatial arrangement of a lower element among its neighbouring elements is dependent on the class of element. For instance, a list item usually follows a bullet in the reading order. Similarly, elements that are part of the same higher-order group tend to be arranged in a spatially co-located pattern. To leverage relative spatial arrangement of all elements in a form together, we arrange them according to a natural reading order (left to right and top to bottom arrangement), encode their context aware representations sequentially using text and spatial coordinates and use them for prediction. For each task, the decoder predicts the output for each element sequentially, conditioning it on the outputs of elements before it in the sequence in an auto-regressive manner (just like sentence generation in NLP). For group extraction task, our model assigns a group id to each element conditioning it on ids predicted for previous elements. This is essential to predict correct group id for current element (for instance, consider assigning same group id to elements that are part of same group).

Let a form be comprising of a list of TextBlocks (f_t) and list of widgets (f_w). We arrange $f_e = f_t \bigcup f_w$ according to natural reading order to obtain arranged sequence a_e which is used as input for both the tasks ('A' in figure 2).

3.1 Element Type Classification

Let t_a and s_a be the list of text content and spatial coordinates (x,y,w,h) corresponding to a_e, where x and y are pixel coordinates from top left corner in image and w & h denote width and height of an element respectively. Our type classification model comprises of three sub-modules namely Text Encoder (TE) which encodes the text representation of each element, Context Encoder (CE) which produces context aware embedding for each element in the sequence, and Type Decoder (TD) which sequentially predicts type output. We discuss each of these modules in detail.

Text Encoder : Consider an element $\{a_e\}_i$ having text $\{t_a\}_i$ comprising of words $\{w_{i1}, w_{i2}, ..., w_{in}\}$. Since the text information is obtained through PDF content, the words often contain noise, making use of standard word vectors difficult. To mitigate this, we obtain word embeddings using python library *chars2vec*[2]. This gives a sequence of embeddings $\{we_{i1}, we_{i2}, ..., we_{in}\}$ which is given as input to an LSTM - TE_{θ_1}, that processes the word embeddings such that the cell state $\{c_t\}_i$ after processing last word is used as text representation for $\{a_e\}_i$ ('B' in figure 2). A widget's textual representation is taken as a vector of 0s.

Context Encoder : Consider a sequence element $\{a_e\}_i$ with corresponding textual representation $\{c_t\}_i$ and spatial coordinates $\{s_a\}_i$. These are concatenated ('C' in figure 2)) together to obtain $\{e\}_i$ representing the element. The sequence e obtained is given as input to a BiLSTM - CE_{θ_2}, which produces a context aware embedding $\{b\}_i$ for each element in the sequence ('D' in figure 2).

Type Decoder : The output from the previous stage is given as input to a final LSTM based decoder - TD_{θ_3}, that sequentially outputs the category type for each element ('F' in figure 2). Specifically, the decoder at time step i is given input

[2]https://github.com/IntuitionEngineeringTeam/chars2vec

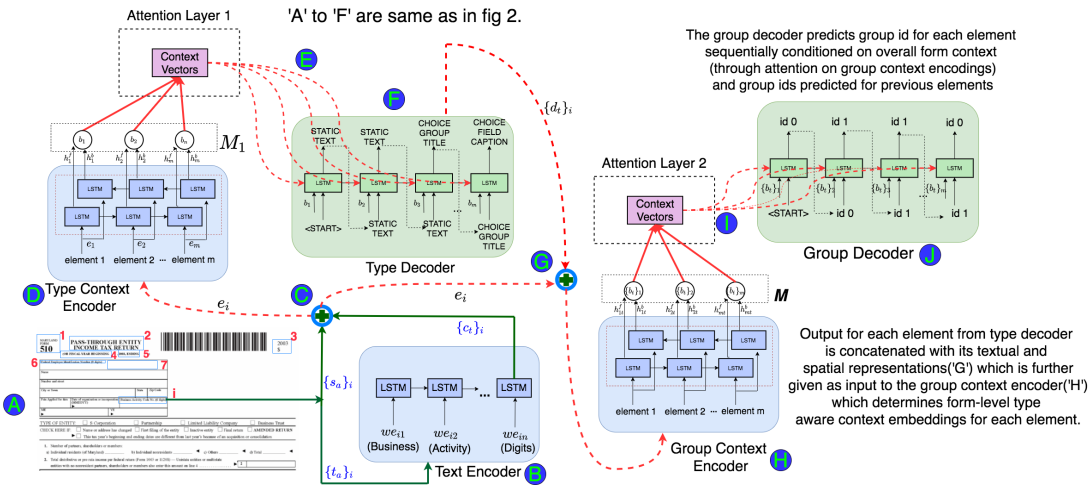

Figure 3: Architecture of our best performing model for group extraction leveraging type model shown in figure 2.

$\{b\}_i$ to predict the type class of i^{th} element. Additionally, we use Bahdnau attention mechanism (Bahdanau et al., 2014) to make TD_{θ_3} attend on context memory M ('E' in figure 2) at each time step of decoding, where M is obtained by stacking $\{b_1; b_2; ..\}$ column-wise. This is to make it easier for decoder to focus on specific elements in sequence while predicting type for current element since elements sequence in a form tends to be very long. A linear layer with softmax activation is used over the decoder outputs for type classification.

We train all 3 modules - TE_{θ_1}, CE_{θ_2} and TD_{θ_3} together using teacher forcing technique (Williams and Zipser, 1989) and standard cross entropy loss.

3.2 Higher Order Group Identification

Our second task is to identify larger groups. Consider one such group - ChoiceGroup, comprising of a collection of TextBlocks and Widgets having different semantics(illustrated in figure 1). A ChoiceGroup contains 1) an optional choice group title which contains details and instructions regarding filling it; and 2) a collection of choice fields which are boolean fields such that each field comprises of a textual caption - choice field caption, and one or more choice field widgets. We formulate target label prediction for this task as that of predicting a cluster/group id for each element. Consider the element sequence a_e such that elements $\{\{a_e\}_{i1}, \{a_e\}_{i2}, ...\}$ are part of a group. We assign this group a unique number and train the model to predict same group number for each of these elements. Elements that are not part of any group are assigned a reserved group i.e. 0.

We adopt a similar model as used for type classification except instead of type decoder, we have Group Decoder (GD_{θ_4}) such that projection layer classifies each element into one of the groups. We hypothesize that category type of elements can be a useful clue for group decoder. To leverage the type information, we study a variant of our model - Cascaded Model, where we have a common text encoder but separate context encoders - CE_T & CE_G, and decoders - TD & GD, for the two tasks. Specifically, given a sequence of elements a_e with combined textual and spatial representations e ('C' in figure 3), we first first feed them into type context encoder (CE_T, 'D' in figure 3) and type decoder (TD, 'F' in figure 3) as before to obtain decoder output sequence d_t for each element. We modify the output types to categories which are relevant to the grouping task - ChoiceGroup Title, TextField Caption, ChoiceField Caption, ChoiceWidget, Text Widget, other TextBlocks. Since an element can be part of a field which is contained in choice group, we use two separate FC layers on decoder output to predict separate group ids for the element while determining choice groups and fields.

TD outputs are concatenated with e for each element ('G' in figure 3) and given as input to group context encoder CE_G to obtain contextual outputs sequence b_t ('H' in figure 3). The group decoder GD ('J' in figure 3) uses the sequence b_t as input and attention memory ('I' in figure 3) during decoding. For d_t, we purposely use outputs of type decoder LSTM and not final type projection layer outputs as determined empirically in experiments section. All five modules - TE, CE_T, TD, CE_G and GD are trained end-to-end for both tasks simultaneously.

Model	Choice Widget	Text Widget	Choice GroupTitle	Choice Caption	TextField Caption	Header Title	Section Title	Bullet	List Item	Static Text	Overall
$DLV3+$	68.24	96.66	57.90	76.28	86.10	83.55	55.43	48.89	75.94	69.37	84.18
$MFCN$	0.0	81.25	0.0	0.0	46.87	69.42	**71.47**	90.03	54.26	11.29	48.59
A_T (ours)	67.77	85.92	56.18	66.81	80.72	82.38	57.20	82.70	81.84	70.24	76.92
B_T (ours)	90.84	**98.26**	76.81	89.55	**91.36**	83.28	57.02	91.58	**90.91**	82.18	88.87
C_T (ours)	**91.83**	96.89	**78.93**	**90.53**	91.27	**85.88**	67.48	**93.55**	90.78	**85.31**	**90.06**

Table 1: Element type classification accuracy of different ablation methods and baselines. Here A_T, B_T and C_T are different Form2Seq variants. A_T gets only element's spatial coordinates as input, B_T gets additional single bit depicting if an element is a TextBlock or a Widget in addition to their spatial coordinates, and C_T gets both textual and spatial information as inputs but does not receive the additional bits provided to B_T.

4 Experiments

4.1 Dataset

Forms Dataset: We have used our Forms Dataset comprising of 23K forms[3] across different domains - automobile, insurance, finance, medical, government (court, military, administration). We employed annotators to mark bounding box of higher order structures in form images as well as lower level constituent elements for each structure. There were multiple rounds of review where we suggested specific cases for each structure and patterns for correction to annotators. We discuss distribution of different structures across (train/test) splits for 10 element types : TextField Caption (129k/31.6k), TextField Widget (222k/533k), Choice Field Caption (35k/8.9k), ChoiceField Widget (39.2k/9.94k), ChoiceGroup Title (8.92k/2.28k), Header Title (10.2k/2.57k), Section Title (28.5k/7.25k), Bullet (56.4k/14.2k), List Item (58.9k/14.7k), Static Text (241.k/61.2k). For higher order structures, distribution of text fields and choice fields is same as that for their captions while for choice groups it is (15.5k/1.76k). Each form was tagged by an annotator(both lower and higher-level structures) and then reviewed by some other annotator. In $\sim$85% forms, no corrections were made but some minor corrections were made in the rest 15% cases after review phase.

ICDAR 2013: We also evaluate our approach on the table structure recognition task on ICDAR 2013 dataset. It comprises of 156 tables from two splits - US and EU set. We extract the images from the pdfs and train our model to extract the table structure by grouping table text into rows and columns. We divide 156 tables into a set of 125 tables for training and 31 for testing following the strategy employed by (Siddiqui et al., 2019) and compare the performance of our approach with them.

[3]Due to legal issues, we cannot release entire dataset. However, the part we plan to release will be large comprising of rich annotations and representative of our entire diverse set. It will be made available at: https://github.com/Form2Seq-Data/Dataset

4.2 Implementation Details

For text encoder TE, we fix size of text in a TextBlock to maximum 200 words. We use chars2vec model which outputs 100 dimensional embedding for each word and fix LSTM size to 100. For type classification, we use a hidden size of 500 for both forward and backward LSTMs in CE_T and a hidden size of 1000 for decoder TD with size of attention layer kept at 500. We tune all hyper-parameters manually based on validation set performance. Final type projection layer classifies each element into one of 10 categories. For grouping task, both isolated and cascaded model have exactly same configuration for CE_G and GD as for type modules. For cascaded model, type projection layer classifies each element into relevant type categories as discussed in Methodology section. We train all models using Adam Optimizer (Kingma and Ba, 2014) at a learning rate of 1×10^{-3} on a single Nvidia 1080Ti GPU. We determined and used largest batch size(=8) that fits memory.

4.3 Results and Discussion

Type Classification : Results for type classification are summarized in Table 1. We compare three models; A_T - where we only give elements' spatial coordinates as input, B_T - where we additionally give a single bit depicting if an element is a TexBlock or a Widget, and C_T where both textual and spatial information is given as input. Using only coordinates yields inferior results since the model only has information regarding arrangement of elements. Adding textblock/widget flag significantly improves the overall accuracy by $\sim$ 12% (A_T to B_T). Adding textual information (model C_T) improves the overall accuracy by 1.19% to

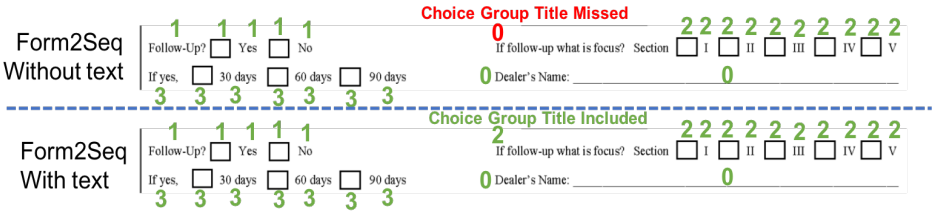

Figure 4: Predictions for a form snippet: Adding text input helps Form2Seq identify title which improves grouping.

90.06%. The accuracy for SectionTitle improves substantially from 57.02% to 67.48% and shows an improvement of 0.99%, 2.12%, 0.98%, 2.6%, 1.97%, 3.13% for ChoiceWidget, ChoiceGroupTitle, ChoiceCaption, HeaderTitle, Bullet and StaticText respectively.

Group Identification : We report precision and recall numbers for the task of group extraction. Segmentation methods commonly use area overlap thresholds such as Intersection over Union(IoU) while matching expected and predicted structures(we evaluate baselines with an IoU threshold of 0.4). For our method, given a set of ground truth groups $\{g_1, g_2, g_3, ..., g_m\}$ and a set of predicted groups $\{p_1, p_2, p_3, ..., p_k\}$, we say a group p_i matches g_j iff the former contains exactly the same TextBlocks and Widgets as the latter. It takes into account all the lower elements which constitute the group (necessary to measure structure extraction performance). Thus, this metric is stricter than IoU based measures with any threshold since a group predicted by our method and evaluated to be correct implies that bounding box of prediction(obtained by taking the union of elements in it) will exactly overlap with expected group.

We first analyse the performance of our method on extracting choice groups. We consider different variants of our approach : 1) model A_G - grouping in isolation; 2) model B_G - both type and grouping task simultaneously with shared context encoder, type decoder attends on context encoder outputs while group decoder attends on context encoder outputs and type decoder outputs separately; 3) model C_G - type identification trained separately, its classification outputs is given as input to group context encoder non-differentiably; 4) model D_G - same as B_G except separate context encoders for two tasks and softmax outputs concatenated with textual and spatial vectors as input to group context encoder; 5) model E_G - same as D_G except instead of softmax outputs, type decoder LSTM outputs are used; and 6) F_G(noText) - same as E_G except spatial coordinates with isText signal used as input.

Model	Recall	Precision	F-Score
$DLV3+$	35.65	57.95	44.14
$MFCN$	16.97	11.86	13.96
A_G(ours)	51.18	55.48	53.24
B_G(ours)	53.18	56.22	54.65
C_G(ours)	55.9	57.15	56.51
D_G(ours)	50.82	54.88	52.77
E_G(ours)	**58.67**	**60.81**	**59.72**
F_G(ours)	55.32	56	55.65

Table 2: Comparison between F-scores of different models and baselines for ChoiceGroup Identification only. A_G to F_G are different variants of Form2Seq.

Table 2 shows joint training of both tasks improves F-score from 53.24 to 54.65 (A_G to B_G) with improvement of 1.86 if type information is incorporated non-differentiably(B_G to C_G). Our best performing model(E_G) achieves an F-score of 59.72. We observe that using type projection layer softmax outputs instead results in poor performance(E_G vs D_G). We observe that using **text** in Form2Seq(E_G) performs 4.07 points better in F-score vs. ablation F_G(w/o text). It can be seen in figure 4 that F_G misses choice group title(red), while Form2Seq with text(E_G) extracts complete choice group[4].

Comparison with baselines : We consider two image semantic segmentation baselines - DeepLabV3+ (DLV3+) (Chen et al., 2018) and MFCN (Yang et al., 2017). For fair comparison, we implement two variants of each baseline - 1) only form image is given as input; 2) textblocks and widgets masks are given as prior inputs with image. We train both variants with an aspect ratio preserving resize of form image to 792x792. For MFCN, loss for different classes are scaled according to pixel area as described in their work. To classify type of an element, we post process prediction masks for

[4]Please refer to supplementary for more visualisations

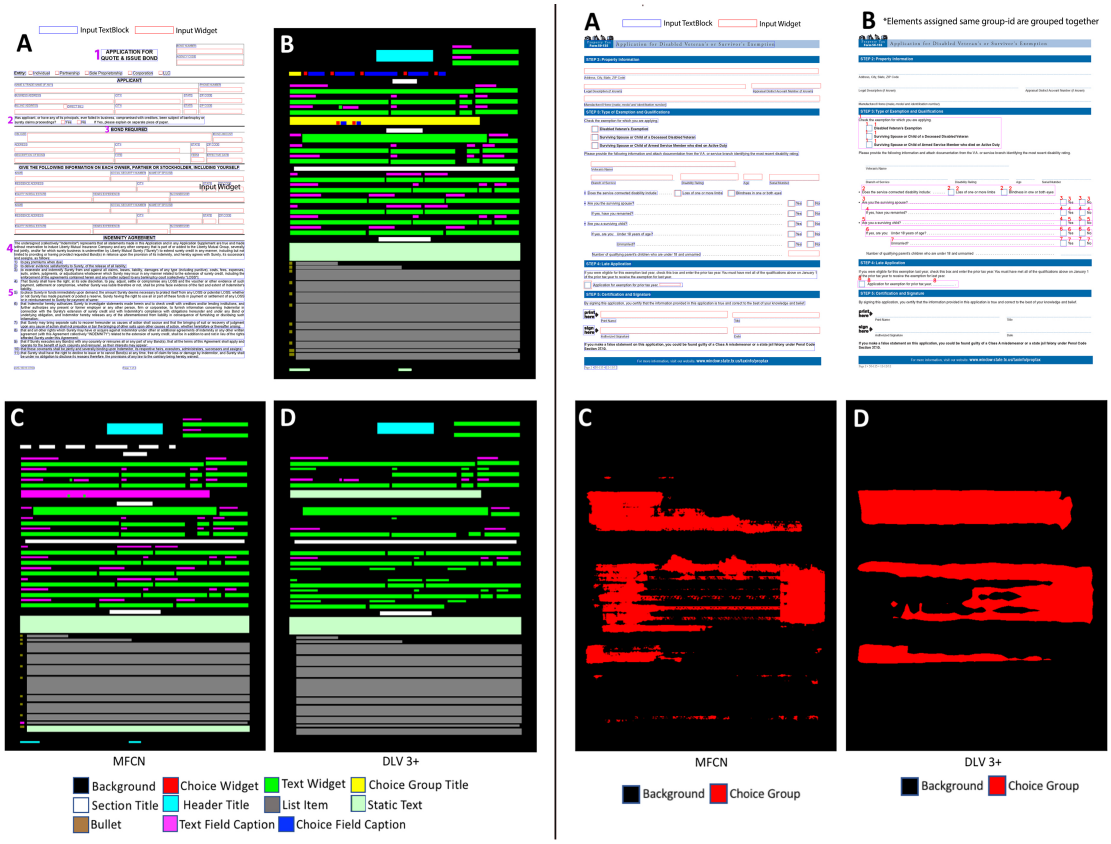

Figure 5: Examples of type classification (left) and choice group extraction (right). Top row shows form (A) and our outputs (B). For type predictions, we visualise our classification outputs as mask for understanding, and show post processed baseline outputs(through majority voting based on predicted masks). We can see that our Form2Seq framework makes better classifications for elements (2,3,5) marked in the top left image (1=Header Title, 2=Choice Group Title, 3=Section Title, 4=Static Text and 5=Bullet). For grouping task, elements highlighted with the same number by our model are predicted as part of same group(zoom in for viewing). Bottom row shows baseline segmentation outputs (C and D).

baselines by performing a majority voting among pixels contained inside it for that particular element. For MFCN, without prior variant performed better, unlike DLV3+. We report metrics corresponding to better variant. As can be seen in table 1, our best performing model (C_T) significantly outperforms both baselines in accuracy. Our model performs better for almost all category types. We observe that DLV3+ and MFCN are not able to perform well for all type classes simultaneously - DLV3+ performs sub-optimally for ChoiceWidget, Bullet and StaticText while MFCN performs poorly for ChoiceWidget, ChoiceGroup Title, Choice Field Caption even after loss scaling. We believe since forms are dense, such methods fail to distinguish different regions and capture complex concepts, for instance MFCN predicts '2'(shown in figure 5 (left)) as text field caption instead of choice group title due to widgets present around it.

For baselines, we match expected groups with segmented outputs through IoU overlap, keeping a threshold (0.40) for determining correct match. Since higher order groups span across different lower elements boundaries, it is not possible to leverage them to refine group masks predicted by baselines. Our proposed model (evaluated with stricter measure) outperforms DLV3+ (better baseline) by 15.58 in F-Score(as can be seen in Table 2), even though it has lesser parameters(31.2 million) than DLV3+(59.4 million). Further, our main model (E_G) when evaluated through IoU overlap threshold of 0.40 achieves even higher recall, precision, F-Score of **74.3**, **78.6** and **76.3** respectively. Figure 5 shows outputs obtained using our approach and baseline methods. For grouping task (right), DLV3+ recognises couple of choice groups correctly but provides incomplete predictions in remaining regions, often merging them owing to its disability to disambiguate groups in dense areas. MFCN could not capture

Construct	DLV3+			MFCN			Ours		
	R	P	F	R	P	F	R	P	F
Text Field	43.64	34.63	38.62	37.12	38.94	38.0	**71.59**	**80.6**	**75.82**
Choice Field	61.93	44.42	51.73	31.45	14.24	19.6	**83.48**	**88.71**	**86.01**
Choice Group	43.25	53.5	47.83	30.99	26.85	28.77	**59.27**	**64.2**	**61.63**

Table 3: Recall(R), Precision(P) and F-score(F) of different methods on extracting different group structures together - text field, choice field and choice group simultaneously.

Model	Table-Rows			Table-Columns			Average		
	P	R	F1	P	R	F1	P	R	F1
Baseline (Siddiqui et al., 2019)	**95.3**	94.2	94.8	91.6	92.6	92.1	93.4	93.4	93.4
Ours	94.2	**96.1**	**95.1**	**95.7**	**92.9**	**94.3**	**95.0**	**94.5**	**94.7**

Table 4: Comparison with baseline on Table Structure Recognition (identifying rows and columns) task on ICDAR-2013 dataset.

horizontal context between Choice group Title and Choice Fields and outputs broken predictions. In comparison, our model extracted 7 out of 8 choice groups correctly.

Extracting Higher Order Constructs Simultaneously: We train our model to detect choice groups, text fields and choice fields together. To enable baseline methods to segment these hierarchical and overlapping structures simultaneously in separate masks, we use separate prediction heads on penultimate layer's output. Table 3 shows the results obtained. Our method works consistently well for all the structures outperforming the baselines.

Table Structure Recognition : We further evaluate our proposed framework on a different task of grouping text in a table into rows and columns on publicly available ICDAR 2013 dataset. The input to our framework is the sequence of texts (arranged in natural reading order as usual) present in a table. We train our model to predict same group id for texts present in the same row and simultaneously detect columns in a similar manner using a separate prediction head. As a post processing step, we consider different sets of texts which are aligned vertically (sharing common horizontal span along the x-axis). We then consider the column group ids predicted by the model and assign majority column id (determined for a set using texts present in it) to all the texts in the set. The re-assigned ids are then used to determine different groups of texts to recognise columns. We perform similar processing while determining the final rows. Siddiqui et al. proposed to perform this task through constrained semantic segmentation achieving state-of-the-art results. Table 4 summarises the results obtained and compares our approach with (Siddiqui et al., 2019) showing our method obtains better F1 score for both rows, columns and average metrics (as used and reported in their paper).

5 Conclusion

We present an NLP based Form2Seq framework for form document structure extraction. Our proposed model uses only lower level elements - textblocks & widgets without using visual modality. We discuss two tasks - element type classification and grouping into larger constructs. We establish improvement in performance through text info and joint training of two tasks. We show that our model performs better compared to current semantic segmentation approaches. Further we also perform table structure recognition (grouping texts present in a table into rows and columns) achieving state-of-the-art results. We are also releasing a part of our forms dataset to aid further research in this direction.

References

Hassan Alam and Fuad Rahman. 2003. Web document manipulation for small screen devices: A review. In *Web Document Analysis Workshop (WDA)*.

Nicolas Audebert, Catherine Herold, Kuider Slimani, and Cédric Vidal. 2020. Multimodal deep networks for text and image-based document classification. In *Machine Learning and Knowledge Discovery in Databases*, pages 427–443, Cham. Springer International Publishing.

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

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2014. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*.

Timo I. Denk and Christian Reisswig. 2019. {BERT}grid: Contextualized embedding for 2d document representation and understanding. In *Workshop on Document Intelligence at NeurIPS 2019*.

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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dimitrios Drivas and Adnan Amin. 1995. Page segmentation and classification utilising a bottom-up approach. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 2, pages 610–614. IEEE.

Łukasz Garncarek, Rafał Powalski, Tomasz Stanisławek, Bartosz Topolski, Piotr Halama, and Filip Graliński. 2020. Lambert: Layout-aware language modeling using bert for information extraction. *arXiv preprint arXiv:2002.08087*.

Filip Graliński, Tomasz Stanisławek, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. 2020. Kleister: A novel task for information extraction involving long documents with complex layout. *arXiv preprint arXiv:2003.02356*.

Aditya Gupta, Anuj Kumar, VN Tripathi, S Tapaswi, et al. 2007. Mobile web: web manipulation for small displays using multi-level hierarchy page segmentation. In *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology*, pages 599–606. ACM.

Jaekyu Ha, Robert M Haralick, and Ihsin T Phillips. 1995a. Document page decomposition by the bounding-box project. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 2, pages 1119–1122. IEEE.

Jaekyu Ha, Robert M Haralick, and Ihsin T Phillips. 1995b. Recursive xy cut using bounding boxes of connected components. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 2, pages 952–955. IEEE.

Leipeng Hao, Liangcai Gao, Xiaohan Yi, and Zhi Tang. 2016. A table detection method for pdf documents based on convolutional neural networks. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 287–292. IEEE.

Dafang He, Scott Cohen, Brian Price, Daniel Kifer, and C Lee Giles. 2017. Multi-scale multi-task fcn for semantic page segmentation and table detection. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 254–261. IEEE.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

G. Jaume, H. Kemal Ekenel, and J. Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 1–6.

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*, pages 4459–4469, Brussels, Belgium. Association for Computational Linguistics.

Mohamed Khemakhem, Axel Herold, and Laurent Romary. 2018. Enhancing usability for automatically structuring digitised dictionaries.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Andreas Kölsch, Ashutosh Mishra, Saurabh Varshneya, Muhammad Zeshan Afzal, and Marcus Liwicki. 2018. Recognizing challenging handwritten annotations with fully convolutional networks. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 25–31. IEEE.

Frank Lebourgeois, Z Bublinski, and H Emptoz. 1992. A fast and efficient method for extracting text paragraphs and graphics from unconstrained documents. In *Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol. II. Conference B: Pattern Recognition Methodology and Systems*, pages 272–276. IEEE.

Xiaojing Liu, Feiyu Gao, Qiong Zhang, and Huasha Zhao. 2019. Graph convolution for multimodal information extraction from visually rich documents. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 32–39, Minneapolis, Minnesota. Association for Computational Linguistics.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Bodhisattwa Prasad Majumder, Navneet Potti, Sandeep Tata, James Bradley 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*, pages 6495–6504.

Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. 2015. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528.

Fuad Rahman and Hassan Alam. 2003. Conversion of pdf documents into html: a case study of document image analysis. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 1, pages 87–91. IEEE.

Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Ronan Sicre, Ahmad Montaser Awal, and Teddy Furon. 2017. Identity documents classification as an image classification problem. In *International Conference on Image Analysis and Processing*, pages 602–613. Springer.

S. A. Siddiqui, P. I. Khan, A. Dengel, and S. Ahmed. 2019. Rethinking semantic segmentation for table structure recognition in documents. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1397–1402.

Noah Siegel, Nicholas Lourie, Russell Power, and Waleed Ammar. 2018. Extracting scientific figures with distantly supervised neural networks. In *Proceedings of the 18th ACM/IEEE on joint conference on digital libraries*, pages 223–232. ACM.

Anikó Simon, J-C Pret, and A Peter Johnson. 1997. A fast algorithm for bottom-up document layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):273–277.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Christoph Wick and Frank Puppe. 2018. Fully convolutional neural networks for page segmentation of historical document images. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 287–292. IEEE.

Curtis Wigington, Chris Tensmeyer, Brian Davis, William Barrett, Brian Price, and Scott Cohen. 2018. Start, follow, read: End-to-end full-page handwriting recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 367–383.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. Layoutlm: Pretraining of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200.

Xiao Yang, Ersin Yumer, Paul Asente, Mike Kraley, Daniel Kifer, and C Lee Giles. 2017. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5315–5324.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.