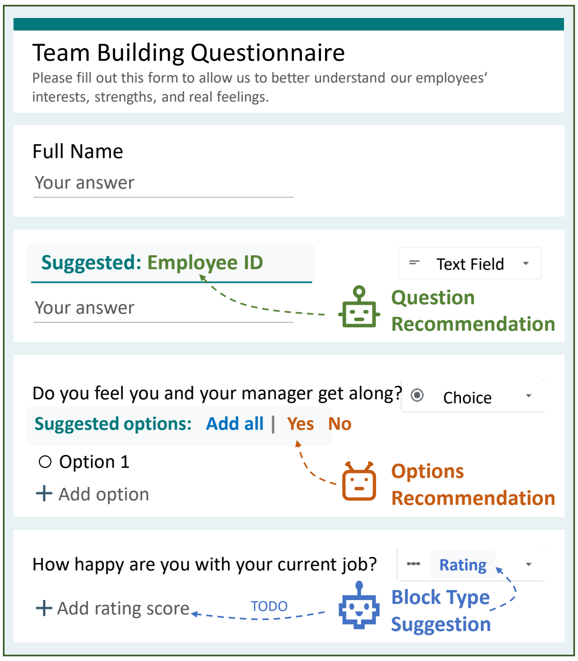
FormLM: Recommending Creation Ideas for Online Forms by Modelling Semantic and Structural Information

Yijia Shao[1*] Mengyu Zhou[2†] Yifan Zhong[3*] Tao Wu[4] Hongwei Han[5*]
Shi Han[2] Gideon Huang[4] Dongmei Zhang[2]
[1]Peking University [2]Microsoft Research [3]Fudan University [4]Microsoft [5]Tsinghua University
shaoyj@pku.edu.cn, {mezho, twu, shihan, gihuang, dongmeiz}@microsoft.com
yfzhong20@fudan.edu.cn, hhw20@mails.tsinghua.edu.cn

Abstract

Online forms are widely used to collect data from human and have a multi-billion market. Many software products provide online services for creating semi-structured forms where questions and descriptions are organized by predefined structures. However, the design and creation process of forms is still tedious and requires expert knowledge. To assist form designers, in this work we present **FormLM** to model online forms (by enhancing pre-trained language model with form structural information) and recommend form creation ideas (including question / options recommendations and block type suggestion). For model training and evaluation, we collect the first public online form dataset with 62K online forms. Experiment results show that FormLM significantly outperforms general-purpose language models on all tasks, with an improvement by 4.71 on Question Recommendation and 10.6 on Block Type Suggestion in terms of ROUGE-1 and Macro-F1, respectively.



Team Building Questionnaire
Please fill out this form to allow us to better understand our employees' interests, strengths, and real feelings.

Full Name
Your answer

Suggested: Employee ID
Your answer

Do you feel you and your manager get along?
Suggested options: Add all | Yes No
○ Option 1
+ Add option

How happy are you with your current job?
+ Add rating score

Figure 1: An Example Online Form with the Three Tasks of Intelligent Form Creation Ideas.

1 Introduction

Online forms are widely used to collect data in everyday scenarios such as feedback gathering (Ilieva et al., 2002), application system (Sylva and Mol, 2009), research surveys (Yarmak, 2017), *etc.* With a multi-billion market (Research and Markets, 2021), many software products – such as Survey Monkey (Abd Halim et al., 2018), Google (Mondal et al., 2018) and Microsoft Forms (Rhodes, 2019) – provide services to help users create online forms which consist of multiple blocks (*e.g.*, Figure 1).

However, there are obstacles preventing the creation of well-designed online forms, which could hurt response rate and quality (Krosnick, 2018). For each form question, form designers need to write an informative title, specify its type, and provide other required components. This process is tedious and time-consuming even for experienced users. Also, non-experts may be unsure about what question to add or which question type to choose. To improve the experience and efficiency of form composing, it is desirable that online form services could recommend creation ideas to form designers.

To address the above demands, in §3 we identify three machine learning (ML) tasks of **Form Creation Ideas**, including *Question Recommendation*, *Block Type Suggestion*, and *Options Recommendation*. For example, in Figure 1, when one adds a text field block as the second block, the Question Recommendation suggests "Employee ID" for the question based on the existing content (form title, description, and the first question "Full Name"). When editing the third choice question block, the Options Recommendation suggests "Yes" and "No"

[*] The contributions by Yijia Shao, Yifan Zhong and Hongwei Han have been conducted and completed during their internships at Microsoft Research Asia, Beijing, China.

[†] Corresponding author.

as candidate options. Finally, if the user types "How happy are you with your current job?" for the fourth block but hasn't selected a block type yet, the Block Type Suggestion predicts it as a rating type block.

The above tasks require a specifically designed model to understand semi-structured forms, where natural language (NL) text is organized by predefined structures. A form is composed of a title, a description, and a series of blocks. For each block, its subcomponents also follow unique structures. For example, a *Choice* block contains a list of options which serve as candidate answers to the question displayed in the block title. Existing pre-trained language models (PLMs) focus on general-purpose free-form NL text (Devlin et al., 2019; Yang et al., 2019). They may provide a good starting point to model the rich semantic information within NL contents of a form. However, they cannot directly handle the extra structural information of the form. *Is it possible to infuse a PLM with structural information of online forms?*

In this paper, we propose **FormLM** to model both the semantic and structural information of online forms. As we will discuss in §4, there are three key parts of FormLM. First, the form serialization procedure, which represents a form as a tree and converts it into a token sequence without information loss. Second, inheriting existing PLM with a small number of additional parameters: FormLM inherits the parameters of BART (Lewis et al., 2020) to leverage its language modelling capabilities. Also, by adding extra biases to the attention layers, FormLM explicitly handles the structural information. Third, continual pre-training with collected online forms: for better downstream application: We propose two structure-aware objectives – Span Masked Language Model and Block Title Permutation – to continually pre-train FormLM on top of the inherited and additional parameters.

We evaluate FormLM on Form Creation Ideas tasks using our **OOF** (**O**pen **O**nline **F**orms) dataset. This dataset (see §2.2) is created by crawling and parsing public forms on the Web. Comparing to PLMs such as BART, FormLM improves the ROUGE-1 score from 32.82 to 37.53 on Question Recommendation, and the Macro-F1 score from 73.3 to 83.9 on Block Type Suggestion.

In summary, our main contributions are:

- We put forward the problem of online form modeling and formally define a group of tasks on Form Creation Ideas. To the best of our knowledge, these problems have not been systematically studied before.
- FormLM is proposed by us to model both the semantic and structural information by enhancing PLM with form serialization, structural attention and continual pre-training.
- The public OOF dataset with 62k forms is constructed by us. To the best of our knowledge, this is the first public online form dataset. OOF dataset, FormLM code and models are also open sourced at `https://github.com/microsoft/FormLM`.
- Comprehensive experiments – especially baseline comparisons, ablation studies, design choices and empirical studies – are designed and run by us to evaluate the effectiveness of FormLM on the tasks of Form Creation Ideas with the form dataset.

2 Preliminaries

In this section, we further elaborate the predefined structure in online forms, and introduce our collected dataset.

2.1 Online Form Structure

Modern online form services usually allow users to create a form by piling up different types of blocks. There are eight common block types: *Text Field, Choice, Time, Date, Likert, Rating, Upload*, and *Description*. Each block type has a predefined structure (*e.g.*, the options of a choice block) and corresponds to a specific layout shown in the user interface (*e.g.*, bullet points or checkboxes of the options). The order of the blocks in a form usually matters because they are designed to organize questions in an easy-to-understand way, and to collect data from various related aspects. For example, in Figure 1, easier profile / fact questions are asked before the preference / opinion questions.

As shown at the top of Figure 3, an online form can be viewed as an ordered tree. The root node T represents the form title, and its children nodes $\mathrm{Ch}(T) = (\mathrm{Desc}, B_1, ..., B_N)$ represent the form description and a series of blocks. The subtree structure of B_i depends on its type. For *Choice* and *Rating* blocks, $\mathrm{Ch}(B_i) = (\mathrm{Type}_i, \mathrm{Title}_i, \mathrm{Desc}_i, C_i^{(1)}, ..., C_i^{(n_i)})$ where $C_i^{(k)}$ are the options or scores; For

Likert (Johns, 2010) blocks, $\mathrm{Ch}(B_i) = (\mathrm{Type}_i, \mathrm{Title}_i, \mathrm{Desc}_i, R_i^{(1)}, ..., R_i^{(m_i)}, C_i^{(1)}, ..., C_i^{(n_i)})$ where $R_i^{(j)}$ are rows and $C_i^{(k)}$ are columns; For the remaining block types, $\mathrm{Ch}(B_i) = (\mathrm{Type}_i, \mathrm{Title}_i, \mathrm{Desc}_i)$. All description parts (Desc) are optional.

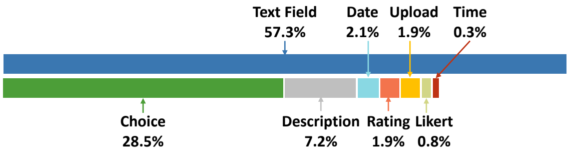


Figure 2: Distribution of Block Types in Online Forms.

2.2 Online Form Dataset

Since there is no existing dataset for online forms, we construct our own OOF (Open Online Forms) dataset by crawling public online forms created on a popular online form website. We filter out forms with low quality and only consider English forms in this work. In total, 62K public forms are collected across different domains, *e.g.*, education, finance, medical, community activities, *etc.*

Due to the semi-structured nature of online forms, we further parsed the crawled HTML pages into JSON format by extracting valid contents and associating each block with its type. Figure 2 shows the distribution of block types in our collected dataset. More details of the dataset construction and its statistics can be found in Appendix A.

3 Form Creation Ideas

As illustrated in Figure 1, when adding a new block, one needs to specify its type and title in the first step. Then, other required components – such as a list of options for a *Choice* block – are added according to the block type. In this paper, we focus on the following three tasks which provide Form Creation Ideas to users in the first and later steps.

Question Recommendation The Question Recommendation aims at providing users with a recommended question based on the selected block type and the previous context. Formally, the model needs to predict Title_i based on T, Desc, $B_1, ..., B_{i-1}$ and Type_i. For example, in Figure 1, it is desirable that the model could recommend "Employee ID" when the form designer creates a *Text Field* block after the first block.

Block Type Suggestion Different from the scenario of Question Recommendation, sometimes form designers may first come up with a block title without clearly specifying its block type. The Block Type Suggestion helps users select a suitable type in this situation. For example, for the last block of Figure 1, the model will predict it as a *Rating* block and suggest adding candidate rating scores if the form designer has not appointed the block type himself / herself. Formally, given Title_i and the available context $(T, \mathrm{Desc}, B_1, ..., B_{i-1})$, the model should predict Type_i in this task.

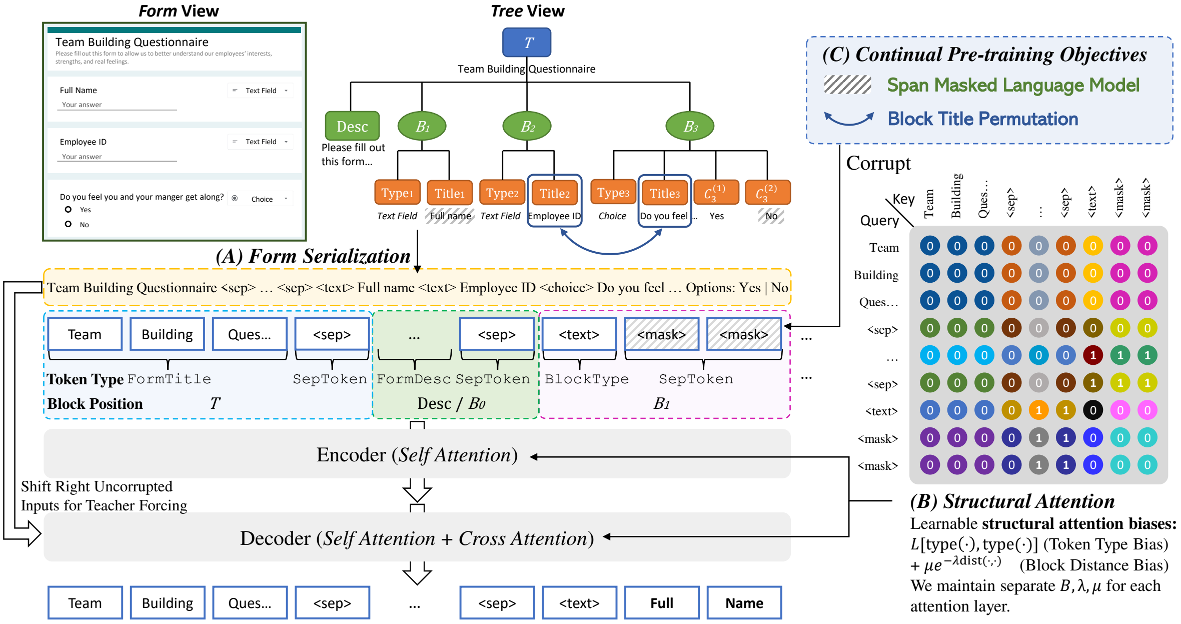
Options Recommendation As Figure 2 shows, *Choice* blocks are frequently used in online forms. When creating a *Choice* block, one should additionally provide a set of options, and the Options Recommendation helps in this case. Given the previous context $(T, \mathrm{Desc}, B_1, ..., B_{i-1})$ and Title_i, the model predicts $C_i^{(1)}, ..., C_i^{(n_i)}$ if $\mathrm{Type}_i = \mathit{Choice}$. In this work, we expect the model to recommend a set of possible options at the same time, so the desired output of this task is $C_i^{(1)}, ..., C_i^{(n_i)}$ concatenated with a vertical bar. For example, in Figure 1, the model may output "Yes | No" to recommend options for the third block.

4 Methodology

As discussed in §1, we propose FormLM to model forms for creation ideas. We select BART as the backbone model of FormLM because it is widely used in NL-related tasks and supports both generation and classification tasks. In the rest of this section, we will describe the design and training details of FormLM as demonstrated in Figure 3.

4.1 Form Serialization

As discussed in §2.1, an online form could be viewed as an ordered tree. In FormLM we serialize the tree into a token sequence which is compatible with the input format of common PLMs. Figure 3(A) depicts the serialization process which utilizes special tokens and separators. First, a special token is introduced for each block type to explicitly encode Type_i. Second, the vertical bar "|" is used to concatenate a list of related items within a block – options / scores $C_i^{(k)}$ of a *Choice* / *Rating* block, and rows $R_i^{(j)}$ or columns $C_i^{(k)}$ of a *Likert* block. Finally, multiple subcomponents of B_i are concatenated using <sep>. Note that there is no information loss in the serialization process, *i.e.*, the hierarchical tree structure of an online form can be reconstructed from the flattened sequence.


Form View
Team Building Questionnaire
Please fill out this form to allow us to better understand our employees' interests, strengths, and real feelings.
Full Name
Text Field
Your answer
Employee ID
Text Field
Your answer
Do you feel you and your manger get along?
Choice
Yes
No
Tree View
T
Team Building Questionnaire
Desc
Please fill out this form...
B1
B2
B3
Type1
Title1
Type2
Title2
Type3
Title3
$C_3^{(1)}$
$C_3^{(2)}$
Text Field
Full name
Text Field
Employee ID
Choice
Do you feel ...
Yes
No
(A) Form Serialization
Team Building Questionnaire <sep> … <sep> <text> Full name <text> Employee ID <choice> Do you feel … Options: Yes | No
Team
Building
Ques...
<sep>
...
<sep>
<text>
<mask>
<mask>
...
Token Type FormTitle
SepToken
FormDesc SepToken
BlockType
SepToken
...
Block Position
T
Desc / B0
B1
Encoder (Self Attention)
Shift Right Uncorrupted
Inputs for Teacher Forcing
Decoder (Self Attention + Cross Attention)
Team
Building
Ques...
<sep>
...
<sep>
<text>
Full
Name
(C) Continual Pre-training Objectives
Span Masked Language Model
Block Title Permutation
Corrupt
Key
Query
(B) Structural Attention
Learnable structural attention biases:
$L[\text{type}(\cdot), \text{type}(\cdot)]$ (Token Type Bias)
$+ \mu e^{-\lambda \text{dist}(\cdot,\cdot)}$ (Block Distance Bias)
We maintain separate B, λ, μ for each attention layer.

Query \ Key	Team	Building	Ques…	<sep>	…	<sep>	<text>	<mask>	<mask>
Team	0	0	0	0	0	0	0	0	0
Building	0	0	0	0	0	0	0	0	0
Ques…	0	0	0	0	0	0	0	0	0
<sep>	0	0	0	0	0	0	0	0	0
…	0	0	0	0	0	0	1	1	1
<sep>	0	0	0	0	0	0	1	1	1
<text>	0	0	0	0	1	1	0	0	0
<mask>	0	0	0	0	1	1	0	0	0
<mask>	0	0	0	0	1	1	0	0	0

Figure 3: The Overview of FormLM Methodology. **(A) Form Serialization** (§4.1) serializes an online form by adding block type tokens and separate tokens to preserve the tree structure. **(B) Structural Attention** (§4.2) encodes the token type and block-level distance by adding structural biases to each attention layer. Different colors in the attention bias matrix denote different items in the lookup table and the number inside each circle represents the block-level distance of a token pair. **(C) Continual Pre-training** (§4.3) requires the model to recover the input sequence corrupted by SpanMLM and BTP. We use the cross-entropy loss between the decoder's output and the uncorrupted sequence for model optimization.

4.2 Structural Attention

Beyond adding structural information into the input sequence, in FormLM we further enhance its backbone PLM with specially designed *Structural Attention* (StructAttn). Our intuition is that the attention calculation among tokens should consider their different roles and locations in a form. *E.g.*, tokens within a question title seldom correlates with the tokens of an option from another question; tokens in nearby blocks (or even the same block) are usually stronger correlated with each other than those from distant blocks.

As illustrated in Figure 3(B), StructAttn encodes the structural information of an online form by adding two bias terms based on the token type (*i.e.*, the role that a token plays in the flattened sequence) and the block-level position. For each attention head, given the query matrix $\mathbf{Q} = [\mathbf{q_1}, \cdots, \mathbf{q_n}]^\top \in \mathbb{R}^{n \times d_k}$, the key matrix $\mathbf{K} = [\mathbf{k_1}, \cdots, \mathbf{k_m}]^\top \in \mathbb{R}^{m \times d_k}$, and the value matrix $\mathbf{V} = [\mathbf{v_1}, \cdots, \mathbf{v_m}]^\top \in \mathbb{R}^{m \times d_v}$, the original output is calculated by

$$\hat{\mathbf{A}} = \frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}, \text{Attn}(H) = \text{softmax}(\hat{\mathbf{A}})\mathbf{V} \quad (1)$$

In FormLM, we add two biases to $\hat{A}$ and the attention head output of StructAttn is calculated by

$$\mathbf{A}_{ij} = \hat{\mathbf{A}}_{ij} + L[\text{type}(\mathbf{q_i}), \text{type}(\mathbf{k_j})] + \mu e^{-\lambda\, \text{d}(\mathbf{q_i}, \mathbf{k_j})}$$
$$\text{Attn}(H) = \text{softmax}(\mathbf{A})\mathbf{V} \quad (2)$$

In Equation (2), the token type bias is calculated based on a learnable lookup table $L[\cdot, \cdot]$ in each attention layer, and the lookup key $\text{type}(\cdot)$ is the type of the corresponding token within the form structure. Specifically, in our work, $\text{type}(\cdot)$ is chosen from 9 token types: `FormTitle`, `FormDesc`, `BlockTitle`, `BlockDesc`, `Option`, `LikertRow`, `LikertColumn`, `BlockType`, `SepToken`. If $\mathbf{Q}$ or $\mathbf{K}$ corresponds to the flattened sequence given by form serialization, $\text{type}(\cdot)$ can be directly obtained from the original form tree; otherwise, in generation tasks, $\mathbf{Q}$ or $\mathbf{K}$ may correspond to the target, and we set $\text{type}(\cdot)$ as the expected output token type, *i.e.*, `BlockTitle` when generating the question and `Option` when generating the options.

Another bias term in Equation (2) is calculated by an exponential decay function to model the relative block-level position, where $\text{d}(\mathbf{q_i}, \mathbf{k_j})$ is the block-level distance between the corresponding tokens of $\mathbf{q_i}$ and $\mathbf{k_j}$ on the form tree. To make $\mathrm{d}(\mathbf{q_i}, \mathbf{k_j})$ well-defined for each token pair, we set Desc as the 0-th block (B_0) and specify $\mathrm{d}(\mathbf{q_i}, \mathbf{k_j})$ as 0 if $\mathrm{type}(\mathbf{q_i})$ or $\mathrm{type}(\mathbf{k_j})$ is equal to `FormTitle`. Note that there are two parameters λ, μ in this term. We make them trainable and constrain their values to be positive to ensure tokens in neighboring blocks give more attention to each other.

We apply StructAttn to three parts of FormLM, self attentions of FormLM encoder, self attentions and cross attentions of FormLM decoder. $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ of encoder self attentions and $\mathbf{K}, \mathbf{V}$ of decoder cross attentions correspond to the source sequence; while $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ of decoder self attentions and $\mathbf{Q}$ of decoder cross attentions correspond to the target sequence. In classification, both the source and the target are the flattened form; while in generation, the target is the recommended question or options.

In §5.5, we will prove the effectiveness of StructAttn through ablation studies and comparing alternative design choices of StructAttn.

4.3 Continual Pre-training

Note that it is difficult to train a model for online forms from scratch due to the limited data. To effectively adapt FormLM to online forms, we conduct continual pre-training on the training set of our collected dataset (see §2.2) with the following two structure-aware objectives.

Span Masked Language Model (SpanMLM) We adapt the masked language model (MLM) to forms by randomly selecting and masking some nodes on the form tree within the masking budget. Compared to SpanBERT (Joshi et al., 2020) which improves the MLM objective by masking a sequence of complete words, we do the masking in a higher level of granularity based on the form structure. Our technique masks a block title, option, *etc.*, instead of arbitrarily masking subword tokens. The latter was proven suboptimal in Joshi et al. (2020); Zhang et al. (2019). Specifically, we use a masking budget of 15% and replacing 80% of the masked tokens with `<MASK>`, 10% with random tokens and 10% with the original tokens.

Block Title Permutation (BTP) As discussed in §2.1, each block can be viewed as a subtree. We introduce the block title permutation objective by permuting block titles in a form and requiring the model to recover the original sequence with the intuition that the model needs to understand the semantic relationship between B_i and $\mathrm{Ch}(B_i)$ to solve this challenge. We randomly shuffle all the block titles to construct the corrupted sequence.

Following the pre-training process of BART, we unify these two objectives by optimizing a reconstruction loss, *i.e.*, we input the sequence corrupted by SpanMLM and BTP and optimize the cross-entropy loss between the decoder's output and the original intact sequence.

5 Experiments

5.1 Evaluation Data and Metrics

We evaluate FormLM and other models on the three tasks of Form Creation Ideas (§3) with our OOF dataset (§2.2). The 62k public forms are split into 49,904 for training, 6,238 for validation, and 6,238 for testing. For each task, random sampling is further performed to construct an experiment dataset. Specifically, for each task, we randomly select no more than 5 samples from a single form to avoid sample bias introduced by those lengthy forms. For Question Recommendation and Block Type Suggestion, each sample corresponds to a block and its previous context (see §3). 239,544, 29,558 and 29,466 samples are selected for training, validation and testing, respectively. For Options Recommendation, each sample corresponds to a *Choice* block with context. 124,994, 15,640 and 15,867 samples are selected for training, validation, and testing.

For Question and Options Recommendations, following the common practice in natural language generation research, we adopt ROUGE[1] (Lin, 2004) scores with the questions/options composed by human as the ground truth. During option recommendation, because the model is expected to recommend a list of options at once, we concatenate options with a vertical bar (described in §4.1) for the comparison of generated results and ground truths. Since it is difficult to have a thorough evaluation of the recommendation quality through the automatic metric, we further include a qualitative study in Appendix D and conduct human evaluations for these two generation tasks (details in Appendix E). For Block Type Suggestion, both accuracy and Macro-F1 are reported to take account of the class imbalance issue.

5.2 Baselines

As there was no existing system or model specifically designed for forms, we compare

[1] We use the Hugging Face implementation to calculate the ROUGE score, https://huggingface.co/metrics/rouge.

	Question Recommendation			Options Recommendation			Block Type Suggestion	
	R1	R2	RL	R1	R2	RL	Macro-F1	Accuracy
RoBERTa	-	-	-	-	-	-	73.7$_{\pm 0.02}$	85.8$_{\pm 0.46}$
GPT-2	22.82$_{\pm 0.22}$	9.71$_{\pm 0.04}$	22.37$_{\pm 0.20}$	17.84$_{\pm 0.10}$	11.38$_{\pm 0.05}$	16.94$_{\pm 0.10}$	74.2$_{\pm 0.16}$	85.6$_{\pm 0.06}$
MarkupLM	-	-	-	-	-	-	79.8$_{\pm 0.27}$	88.6$_{\pm 0.13}$
BART$_{\text{BASE}}$	31.48$_{\pm 0.16}$	15.89$_{\pm 0.18}$	30.91$_{\pm 0.16}$	43.53$_{\pm 0.32}$	31.81$_{\pm 0.21}$	41.5$_{\pm 0.29}$	73.4$_{\pm 0.31}$	85.6$_{\pm 0.17}$
BART	32.82$_{\pm 0.05}$	17.06$_{\pm 0.20}$	32.18$_{\pm 0.05}$	46.12$_{\pm 0.12}$	33.74$_{\pm 0.08}$	43.85$_{\pm 0.12}$	73.3$_{\pm 0.28}$	85.3$_{\pm 0.08}$
FormLM$_{\text{BASE}}$	35.9$_{\pm 0.08}$	18.27$_{\pm 0.10}$	35.23$_{\pm 0.04}$	44.14$_{\pm 0.06}$	32.39$_{\pm 0.16}$	42.21$_{\pm 0.10}$	83.0$_{\pm 0.06}$	90.7$_{\pm 0.09}$
↑ BART$_{\text{BASE}}$	4.42	2.38	4.32	0.61	0.58	0.71	9.6	5.1
FormLM	**37.53**$_{\pm 0.07}$	**19.70**$_{\pm 0.15}$	**36.78**$_{\pm 0.12}$	**47.24**$_{\pm 0.02}$	**34.65**$_{\pm 0.14}$	**44.91**$_{\pm 0.08}$	**83.9**$_{\pm 0.11}$	**91.0**$_{\pm 0.08}$
↑ BART	4.71	2.64	4.6	1.12	0.91	1.06	10.6	5.7

Table 1: Results of FormLM and the Baseline Models on the Tasks of Form Creation Ideas. Note that RoBERTa and MarkupLM are encoder-only models, thus cannot be directly applied to generation tasks. We leave their results blank for Question and Options Recommendations where ROUGE scores (R1, R2, RL) are used to evaluate these two generation tasks. Both the averaged metric and its standard deviation (as subscript) are reported for each result over 3 runs. The two gray rows (with up arrow ↑) show the improvement of FormLM over its backbone model.

FormLM with three general-purposed PLMs – RoBERTa (Liu et al., 2020), GPT-2 (Radford et al., 2019) and BART (Lewis et al., 2020), which represent widely-used encoder, decoder, encoder-decoder based models, respectively. To construct inputs for these PLMs, we concatenate NL sentences in the available context (see §3).

MarkupLM (Li et al., 2022), a recent model for web page modeling, is also chosen as a baseline since forms can be displayed as HTML pages on the Internet. To keep accordance with the original inputs of MakupLM, we remove the tags without NL text (*e.g.*, `<script>`, `<style>`) in the HTML file in OOF dataset.

The number of parameters of each model can be found in Appendix B.

5.3 FormLM Implementation

We implement FormLM using the Transformers library (Wolf et al., 2020). FormLM and FormLM$_{\text{BASE}}$ are based on the architecture and parameters of BART[2] and BART$_{\text{BASE}}$[3] respectively.

For continual pre-training, we train FormLM for 15k steps on 8 NVIDIA V100 GPUs with the total batch size of 32 using the training set of the OOF dataset. For all the three tasks of Forms Creation Ideas, we fine-tune FormLM and all baseline models for 5 epochs with the total batch size of 32 and the learning rate of 5e-5. More pre-training and fine-tuning details are described in Appendix C.

In the rest of this paper, each experiment with randomness is run for 3 times and reported with averaged evaluation metrics.

5.4 Main Results

For FormLM and the baseline models (see §5.2), Table 1 shows the results on the Form Creation Ideas tasks. FormLM significantly outperforms the baselines on all tasks.

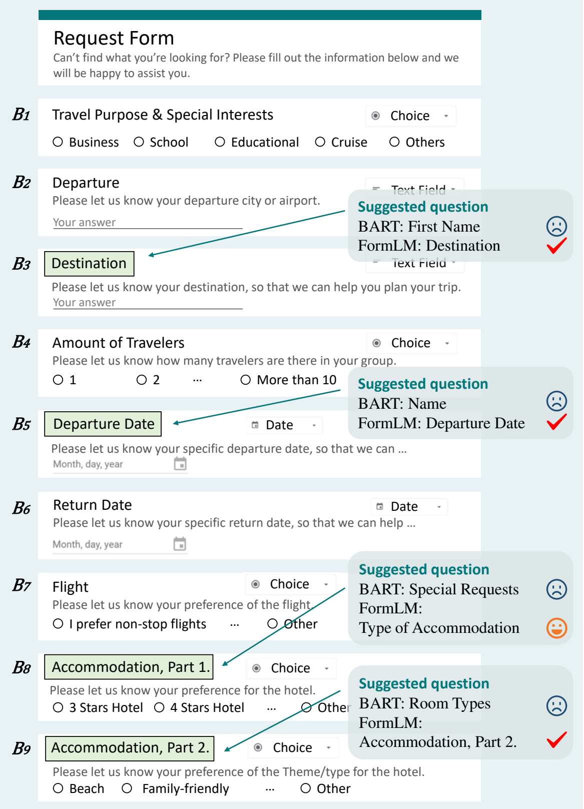
Compared to its backbone BART model (well-known for conditional generation tasks), FormLM further improves the ROUGE-1 scores by 4.71 and 1.12 on Question and Options Recommendations. Human evaluation results in Appendix E also confirm the superiority of FormLM over other baseline models in these two generation tasks. Figure 4 shows questions recommended by BART and FormLM on an example form from the test set. FormLM's recommendations (*e.g.*, "Destination", "Departure Date") are more specific and more relevant to the topic of this form, while BART's recommendations (*e.g.*, "Name", Special Requests") are rather general. Also, after users create B_1, B_2, B_3, B_4 and select B_5 as a *Date* type block, FormLM recommends "Departure Date" while BART recommends "Name" which is obviously not suitable to B_5.

On Block Type Suggestion, FormLM improves the Macro-F1 score by 10.6. The improvement of FormLM over BART (↑ rows in Table 1) shows that our method is highly effective. We will further analyze this in §5.5.

Note that MarkupLM is a very strong baseline

[2] `https://huggingface.co/facebook/bart-large`

[3] `https://huggingface.co/facebook/bart-base`



Request Form

Can't find what you're looking for? Please fill out the information below and we will be happy to assist you.

B1 Travel Purpose & Special Interests ● Choice

○ Business ○ School ○ Educational ○ Cruise ○ Others

B2 Departure Text Field

Please let us know your departure city or airport.

Your answer

B3 Destination Text Field

Please let us know your destination, so that we can help you plan your trip.

Your answer

Suggested question
BART: First Name ☹
FormLM: Destination ✓

B4 Amount of Travelers ● Choice

Please let us know how many travelers are there in your group.

○ 1 ○ 2 ... ○ More than 10

B5 Departure Date Date

Please let us know your specific departure date, so that we can ...

Month, day, year

Suggested question
BART: Name ☹
FormLM: Departure Date ✓

B6 Return Date Date

Please let us know your specific return date, so that we can help ...

Month, day, year

B7 Flight ● Choice

Please let us know your preference of the flight.

○ I prefer non-stop flights ... ○ Other

Suggested question
BART: Special Requests ☹
FormLM:
Type of Accommodation ☺

B8 Accommodation, Part 1. ● Choice

Please let us know your preference for the hotel.

○ 3 Stars Hotel ○ 4 Stars Hotel ... ○ Other

Suggested question
BART: Room Types ☹
FormLM:
Accommodation, Part 2. ✓

B9 Accommodation, Part 2. ● Choice

Please let us know your preference of the Theme/type for the hotel.

○ Beach ○ Family-friendly ... ○ Other

Figure 4: Sample Outputs by FormLM and BART for Question Recommendation. FormLM's recommended questions are more relevant to the topic and more suitable to the selected block type.

for Block Type Suggestion. This model can partly capture the structural information by parsing the form as a DOM (Wood et al., 1998) tree. However, since MarkupLM is not specifically designed for online forms, it is still 4.1 points worse in Macro-F1 than FormLM on this task.

5.5 Analysis of FormLM Designs

	Question	**Options**	**Type**
	R2	R2	F1
Full Model	**19.70**	**34.65**	**83.9**
– Decoder StructAttn	18.90	34.36	83.7
– Encoder StructAttn	19.58	34.41	77.9
– Form Serialization	17.43	33.83	75.5
– Previous Context	12.67	27.65	71.8

Table 2: Ablation Studies on Form Serialization and Structural Attention. "–" means the corresponding component is sequentially removed from FormLM. "– Previous Context" means that the closest block title is the only input.

To further investigate the effectiveness of the design choices in FormLM, we conduct ablation studies and controlled experiments (which are fine-tuned under the same settings as described in §5.3) on the following aspects.

	Question	**Options**	**Type**
	R2	R2	F1
w/o Type Info	17.96	33.97	81.5
w/ Type Info	**19.70**	**34.65**	**83.9**

Table 3: Performance of FormLM "w/" and "w/o" Incorporating the Block Type Information.

Form Serialization For Form Creation Ideas, it is important to model the complete form context (defined in §3). Row "– Previous Context" of Table 2 shows that there is a large performance drop on all the tasks if block title is the only input.[4]

Therefore, we also study the effect of form serialization (see §4.1) which flattens the form context while preserving its tree structure. A naive way of serialization is directly concatenating all available text as NL inputs. Results in this setting (row "– Form Serialization" of Table 2) are much worse than the results of FormLM with form serialization technique. On Block Type Suggestion, the gap is as large as 8.4 on Macro-F1.

Block Type Information A unique characteristic of online forms is the existence of block type (see §2.1). To examine whether FormLM can leverage the important block type information, we run a controlled experiment where block type tokens are replaced by with a placeholder token <type> during form serialization (while other tokens are untouched). As shown in Table 3, removing block type tokens hurts the model performance on all three tasks, which suggests that FormLM can effectively exploit such information.

Structural Attention FormLM enhances its backbone PLM with StructAttn (§4.2). As the row "– Encoder StructAttn" of Table 2 shows, when we ablate StructAttn from FormLM, the Macro-F1 score of Block Type Suggestion drops from 83.9 to 77.9 and the performance on the generation tasks also drops. In FormLM, we apply StructAttn to both encoder and decoder parts. We compare it with the setting without modifying the decoder (row "– Decoder StructAttn") and find applying StructAttn to both the encoder and decoder yields uniformly better results, which may be due to better alignment between the encoder and decoder.

[4] For ablation studies in Table 2, the components are sequentially removed because StructAttn depends on the tree structure preserved in form serialization and both techniques become meaningless if we don't model the form context.

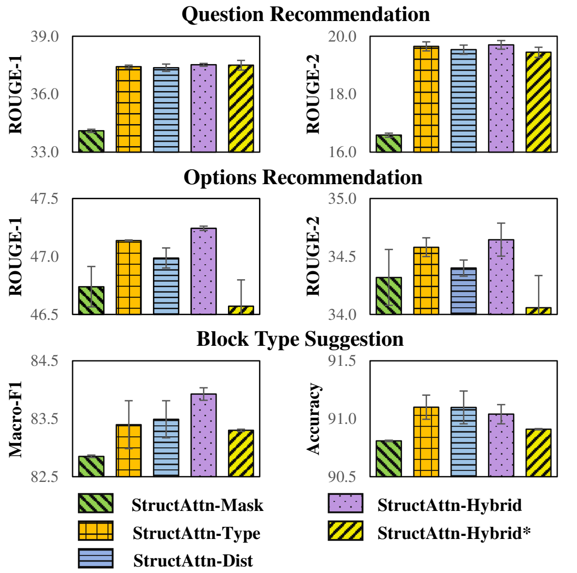


Figure 5: Results of FormLM Using Different Design Choices of StructAttn. (Averaged over 3 runs with std.)

	Question R2	Options R2	Type F1
w/o Pre-training	18.82	33.78	82.2
BTP	19.35	34.18	83.3
SpanMLM	19.42	33.94	83.3
SpanMLM + BTP	**19.70**	**34.65**	**83.9**

Table 4: Ablation Study of Different Continual Pre-training Objectives. (Averaged over 3 runs.)

There are alternative design choices of StructAttn for us to experiment. As Equation (2) shows, there are two bias terms to model the token type and the block-level distance. We compare this design choice ("Hybrid" in Figure 5) with adding only the token type bias ("Type") and only the distance bias ("Dist"). Note that "Hybrid" encodes block-level distance through the exponential decay function, we also compare it with another intuitive design ("Hybrid*") where we use a learnable bias to indicate whether two tokens are within the same block. Besides adding biases, another common practice of modifying attentions is masking. We experiment this design choice ("Mask") by restricting attentions to those tokens in the same node or parent and grandparent nodes within the tree structure. The comparison results are demonstrated in Figure 5. "Mask" performs uniformly worse than adding biases. Among the rest of design choices, "Hybrid" shows slightly better performance on Options Recommendation and Block Type Suggestion.

Continual Pre-training Objectives We design two objectives (§4.3), SpanMLM and BTP, to continually pre-train FormLM on OOF dataset for better domain adaptation. Table 4 shows the ablation results of different objectives. We find FormLM trained with both SpanMLM and BTP performs the best. This suggests SpanMLM which focuses more on the recovery of a single node on the tree and BTP which focuses more on the relationship between different nodes can complement each other.

6 Related Work

(Semi-)Structured Data Modeling In this paper, we mainly focus on modelling parsed form data. They follow well-defined structure and are usually created by software such as online services mentioned in §1. Existing works (Wang et al., 2022a; Xu et al., 2021; Li et al., 2021; Appalaraju et al., 2021; Aggarwal et al., 2020; He et al., 2017) focus on another type of forms, scanned forms (*e.g.*, photos and scanned PDF files of receipts or surveys), and process multi-modal inputs (text, image). This type of forms requires digitization and parsing before passing to any downstream tasks, which are very different from forms studied in this paper.

To the best of our knowledge, the modelling of parsed forms has not been studied before. Existing (semi-)structured data modelling works mainly focus on tables (Yin et al., 2020; Wang et al., 2021), documents (Wan et al., 2021; Liu and Lapata, 2019; Wang et al., 2019), web pages (Wang et al., 2022b), *etc.* Some works represent the (semi-)structured data as a graph and use graph neural network (GNN) for structural encoding (Wang et al., 2020; Cai et al., 2021). Some other works convert (semi-)structured data into NL inputs to directly use PLMs (Gong et al., 2020) or modify a certain part of transformer models – *e.g.*, embedding layers (Herzig et al., 2020), attention layers (Eisenschlos et al., 2021; Yang et al., 2022), the encoder architecture (Iida et al., 2021). Although it is possible to convert online forms to HTML pages to use models like MarkupLM (Li et al., 2022), the results are suboptimal as shown in §5.4 because the unique structural information of online forms are not fully utilized.

Intermediate Pre-training In §4.3 we discussed in FormLM how we adapt a general PLM to the form domain through continual pre-training. Intermediate pre-training of a PLM on the target data (usually in a self-supervised way) has been shown efficient on bridging the gap between PLMs and target tasks (Gururangan et al., 2020; Rongali et al., 2020). Many domain specific models (Xu et al., 2019; Chakrabarty et al., 2019; Lee et al., 2020), including those for (semi-)structured data (Yin et al., 2020; Liu et al., 2022), are built with this technique. Following the previous approaches, we design form-specific structure-aware training objectives for the continual pre-training process.

7 Conclusion

In this paper, we present FormLM for online form modeling. FormLM jointly consider the semantic and structural information by leveraging the PLM and designing form serialization and structural attention. Furthermore, we continually pre-train FormLM on our collected data with structure-aware objectives for better domain adaptation. An extensive set of experiments show that FormLM outperforms baselines on Form Creation Ideas tasks which assist users in the form creation stage.

Limitations

In this work, we conduct research on online form modeling for the first time. While effective in the proposed tasks of Form Creation Ideas, FormLM has some limitations. First, FormLM is designed to assist form designers by recommending questions / options and suggesting the block type. We believe there are more to explore in recommending creation ideas and we plan to design more tasks for Form Creation Ideas, like recommending a whole block, auto-completion, *etc.*, to fully exploit FormLM in the form creation stage. Also, since FormLM performs exceptionally well on Block Type Suggestion, it is worthwhile to consider more fine-grained block types. Second, FormLM only models the form content and leaves out the collected responses. Although form content itself is very informative, it is an important research direction to jointly model online forms and their collected responses for they are useful to other stages of the online form life cycle, especially the form analyzing stage. Furthermore, our collected OOF dataset is limited to English forms and doesn't have manual labels. We hope to enlarge our dataset with non-English forms and investigate the possibility of adding supervised labels to this dataset in the future to further facilitate the study of online forms.

Ethics Statement

Datasets In this work, we collect the public OOF dataset for the research community to facilitate future study of online forms. We believe there is no privacy issue related to this dataset. First, the data sources are public available on the Internet, and are anonymously accessible. We complied with the Robots Exclusion Standard during the data collection stage. Second, our dataset only contains form contents and there are no responses or personal information involved. A checklist has been completed at the researchers' institution to ensure the collected dataset does not have ethical issues.

Risks and Limitations Our work proposes FormLM to model online forms and recommend creation ideas to users in the form designing stage. FormLM uses a pre-trained language model, BART, as the backbone. PLMs have a number of ethical concerns in general, like generating biased or discriminative text (Weidinger et al., 2021) and involving lots of computing power in pre-training or fine-tuning (Strubell et al., 2019). The primary risk of our work is that we formulated Question Recommendation and Options Recommendation as generation tasks, but did not include the post-processing of the generated texts in our pipeline. We suggest post-processing the outputs of FormLM to sift out biased or discriminative text before recommending them to the users when applying our technique to online form services. Designing good post-processing technique is also an interesting avenue for future work.

Another limitation we see from an ethical point of view is that we only consider online forms which use English as the primary language. We are trying to collect online forms in other languages and leave it as a future work to provide a multilingual version of FormLM to assist more users in different parts of the world.

Computational Resources The experiments in our paper require computational resources. However, compared with other LMs pretrained from scratch, FormLM inherits the parameters of its backbone and is continually pre-trained with only 50K online forms. It takes around 8 hours to complete the continual pre-training with 8 NVIDIA V100 GPUs. Despite this, we recognize that not all researchers have access to this resource level, and these computational resources require energy. Notably, all GPU clusters within our organization are shared, and their carbon footprints are monitored in real-time. Our organization is also consistently upgrading our data centers in order to reduce the energy use.

References

Maisaarah Abd Halim, Cik Feresa Mohd Foozy, Isredza Rahmi, and Aida Mustapha. 2018. A review of live survey application: Surveymonkey and surveygizmo. *JOIV: International Journal on Informatics Visualization*, 2(4-2):309–312.

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

Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. 2021. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 993–1003.

Ruichu Cai, Jinjie Yuan, Boyan Xu, and Zhifeng Hao. 2021. Sadga: Structure-aware dual graph aggregation network for text-to-sql. *Advances in Neural Information Processing Systems*, 34:7664–7676.

Tuhin Chakrabarty, Christopher Hidey, and Kathy McKeown. 2019. IMHO fine-tuning improves claim detection. 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 558–563, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Julian Eisenschlos, Maharshi Gor, Thomas Müller, and William Cohen. 2021. MATE: Multi-view attention for table transformer efficiency. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7606–7619, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Heng Gong, Yawei Sun, Xiaocheng Feng, Bing Qin, Wei Bi, Xiaojiang Liu, and Ting Liu. 2020. TableGPT: Few-shot table-to-text generation with table structure reconstruction and content matching. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1978–1988, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

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.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. 2021. TABBIE: Pretrained representations of tabular data. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3446–3456, Online. Association for Computational Linguistics.

Janet Ilieva, Steve Baron, and Nigel M Healey. 2002. Online surveys in marketing research. *International Journal of Market Research*, 44(3):1–14.

Rob Johns. 2010. Likert items and scales. *Survey question bank: Methods fact sheet*, 1(1):11.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Jon A Krosnick. 2018. Questionnaire design. In *The Palgrave handbook of survey research*, pages 439–455. Springer.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chenliang Li, Bin Bi, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. 2021. StructuralLM: Structural pre-training for form understanding. In

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 6309–6318, Online. Association for Computational Linguistics.

Junlong Li, Yiheng Xu, Lei Cui, and Furu Wei. 2022. MarkupLM: Pre-training of text and markup language for visually rich document understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6078–6087, Dublin, Ireland. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022. TAPEX: Table pre-training via learning a neural SQL executor. In *International Conference on Learning Representations*.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Himel Mondal, Shaikat Mondal, Tania Ghosal, and Sarika Mondal. 2018. Using google forms for medical survey: A technical note. *Int J Clin Exp Physiol*, 5(4):216–218.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Urša Reja, Katja Lozar Manfreda, Valentina Hlebec, and Vasja Vehovar. 2003. Open-ended vs. close-ended questions in web questionnaires. *Developments in applied statistics*, 19(1):159–177.

Research and Markets. 2021. Global online survey software market research report (2021 to 2026) - by industry and region. Accessed: 2022-06-14.

Jeffrey M Rhodes. 2019. Creating a survey solution with microsoft forms, flow, sharepoint, and power bi. In *Creating Business Applications with Office 365*, pages 99–103. Springer.

Subendhu Rongali, Abhyuday Jagannatha, Bhanu Pratap Singh Rawat, and Hong Yu. 2020. Continual domain-tuning for pretrained language models. *arXiv preprint arXiv:2004.02288*.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Hella Sylva and Stefan T Mol. 2009. E-recruitment: A study into applicant perceptions of an online application system. *International Journal of Selection and Assessment*, 17(3):311–323.

Hui Wan, Song Feng, Chulaka Gunasekara, Siva Sankalp Patel, Sachindra Joshi, and Luis Lastras. 2021. Does structure matter? encoding documents for machine reading comprehension. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4626–4634, Online. Association for Computational Linguistics.

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2020. RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7567–7578, Online. Association for Computational Linguistics.

Jiapeng Wang, Lianwen Jin, and Kai Ding. 2022a. LiLT: A simple yet effective language-independent layout transformer for structured document understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7747–7757, Dublin, Ireland. Association for Computational Linguistics.

Jingwen Wang, Hao Zhang, Cheng Zhang, Wenjing Yang, Liqun Shao, and Jie Wang. 2019. An effective scheme for generating an overview report over a very large corpus of documents. In *Proceedings of the ACM Symposium on Document Engineering 2019*, DocEng '19, New York, NY, USA. Association for Computing Machinery.

Qifan Wang, Yi Fang, Anirudh Ravula, Fuli Feng, Xiaojun Quan, and Dongfang Liu. 2022b. Webformer: The web-page transformer for structure information extraction. In *Proceedings of the ACM Web Conference 2022*, pages 3124–3133.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1780–1790.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Lauren Wood, Arnaud Le Hors, Vidur Apparao, Steve Byrne, Mike Champion, Scott Isaacs, Ian Jacobs, Gavin Nicol, Jonathan Robie, Robert Sutor, et al. 1998. Document object model (dom) level 1 specification. *W3C recommendation*, 1.

Robert F Woolson. 2007. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials*, pages 1–3.

Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. 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 2324–2335, Minneapolis, Minnesota. Association for Computational Linguistics.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2021. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2579–2591, Online. Association for Computational Linguistics.

Jingfeng Yang, Aditya Gupta, Shyam Upadhyay, Luheng He, Rahul Goel, and Shachi Paul. 2022. TableFormer: Robust transformer modeling for table-text encoding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 528–537, Dublin, Ireland. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Olga Yarmak. 2017. Online surveys in sociology: Opportunities, drawbacks and limitations. In *11th International Conference on Computer Science and Information Technologies CSIT*, volume 4, pages 476–477.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

A Details of Open Online Forms Dataset

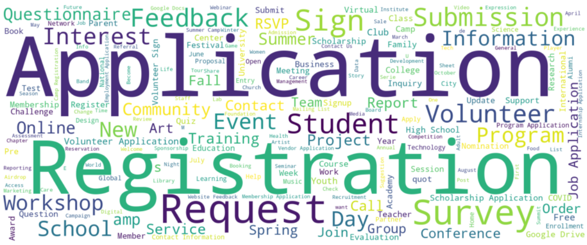


Figure 6: Frequent Words Among Titles of Forms in OOF Dataset.

OOF (Open Online Forms) dataset consists of 62K public forms collected on the Web, covering a wide range of domains and purposes. Figure 6 shows some frequent words among titles of the collected data.

A.1 Dataset Preprocessing

We crawled 232,758 forms created by a popular online form service on the Internet and filter the crawled data using the following constraints: (1) have at least one question block; (2) have no duplicate question blocks; (3) detected as "en"[5] by Language Detection API of Azure Cognitive Service for Language[6]. Finally, 62,380 forms meet all constraints. We randomly split them into 49,904 for training, 6,238 for validation and 6,238 for training.

As introduced in §2.2, we parsed the crawled HTML pages into JSON format according to the online form structure. Specifically, each JSON file contains keys of "title", "description" and "body" which correspond to form title (T), form description ($Desc$), and an array of blocks ($\{B_1, \cdots, B_n\}$). Each block contains keys of "title", "description" and "type". For *Choice* type blocks and *Rating* type blocks, they further contain the key of "options"; for *Likert* type blocks, they further contain keys of "rows" and "columns". For *Description* block, we only keep the plain NL text and remove possible information of other modalities (*i.e*, image, video) because only around 0.1% of *Description* blocks contain video and 2.0% contain image. When parsing the HTML pages into JSON format, we also remove non-ASCII characters within the form.

A.2 Form Length Distribution

We define the length of an online form as the number of blocks within it. Around 80% of collected forms have a form length no greater than 20. The detailed distribution of form length is shown in Figure 7. As we have discussed in §5.1, we further perform random sampling to construct our experiment dataset to avoid sample biases introduced by those lengthy forms.

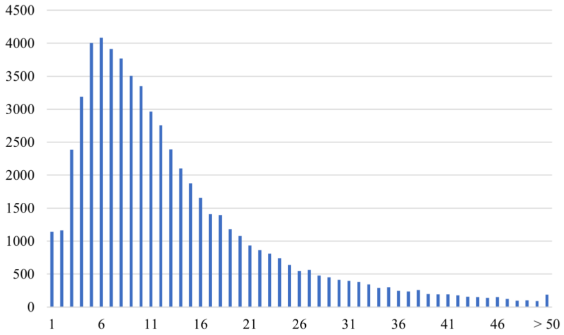


Figure 7: Form Length Distribution of Forms in OOF Dataset.

B Model Configurations

We compare FormLM with four baseline models, RoBERTa, GPT-2, MarkupLM, and BART. FormLM adds a small number of additional parameters to its backbone model (278K for FormLM and 208K for FormLM$_{\text{BASE}}$) to encode structural information in attention layers (§4.2). Table 5 shows model configurations of FormLM and baselines in our experiments.

Model	#Params	#Layers
RoBERTa	124M	12
GPT-2	124M	12
MarkupLM	135M	12
BART$_{\text{BASE}}$	139M	6+6
BART	406M	12+12
FormLM$_{\text{BASE}}$	139M	6+6
FormLM	406M	12+12

Table 5: Model Configurations of FormLM and Baselines.

C More Implementation Details

Continual Pre-training Details We conduct continual pre-training on the training set of the OOF dataset using SpanMLM and BTP objectives (§4.3). We adopt a masking budget of 15% in SpanMLM and do BTP on all training samples. We train

[5] https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes

[6] https://docs.microsoft.com/en-us/azure/cognitive-services/language-service/language-detection/overview

FormLM for 15K steps on 8 NVIDIA V100 GPUs with 32G GPU memory. We set the total batch size as 32 and the max sequence length as 512. We use AdamW optimizer (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and the learning rate of 5e-5. It takes around 8 hours to complete the continual pre-training on our machine.

Fine-tuning Details Among our downstream tasks, Next Question Recommendation and Options Recommendation are formulated as conditional generation tasks. We use the form serialization procedure (§4.1) to convert the available context into model inputs. We fine-tune FormLM for 5 epochs with the total batch size of 32, the max source sequence length of 512, and the max target sequence length of 64. We load the best model which has the highest ROUGE-2 score on the validation set in the training process. During generation, we do beam search and set the beam size as 5. Block Type Classification is formulated as a sequence classification task. We follow the original implementation of BART by feeding the same input into the encoder and decoder and passing the final hidden state of the last decoded token into a multi-class linear classifier for classification. We fine-tune FormLM with 5 epochs with the total batch size as 32 and load the best model which has the highest Macro-F1 score on the validation set during the fine-tuning process.

D Qualitative Study

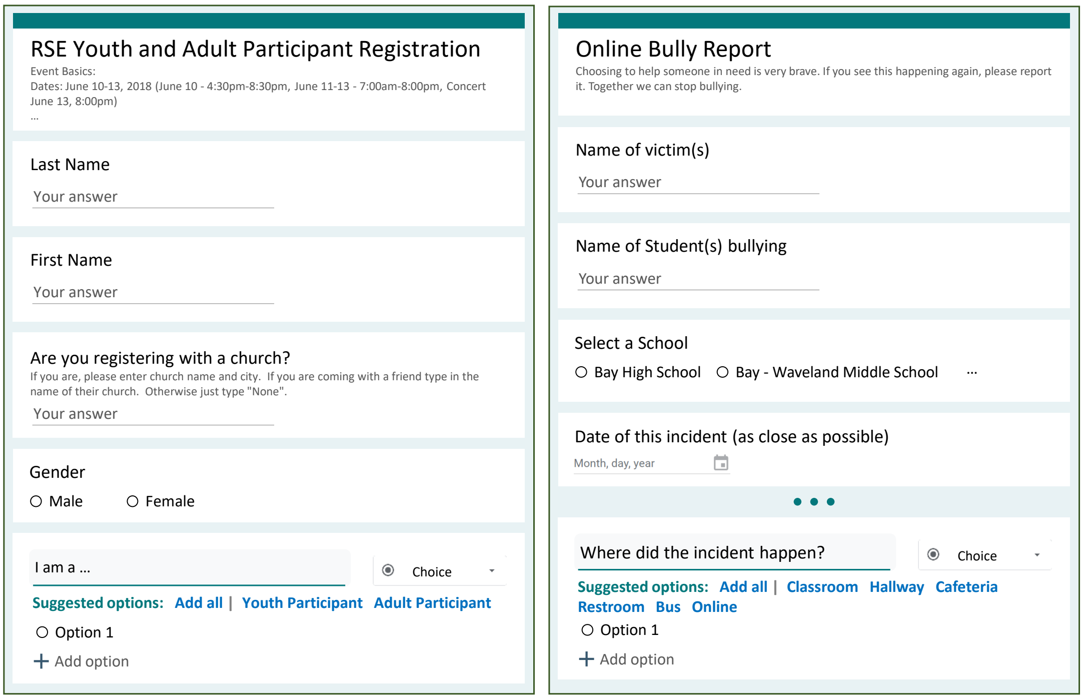
Online forms, as a special format of questionnaires, are mainly used to collect information, *i.e.*, demographic information, needs, preferences, *etc.* (Krosnick, 2018). As shown in Figure 6, the online forms in the OOF dataset are more about objective topics like "Application" and "Registration" because these information collection scenarios prevail in the daily usage. To collect information effectively, a good questionnaire should include questions related to the topic and these questions must be logically connected with each other. Also, for those close-ended questions (the majority of them are *Choice* type questions), they are expected to offer all possible answers for respondents to choose from but not include off-topic options which may cause confusion (Reja et al., 2003). These criteria of good questionnaires restrict the searching space of online form composition, thus making the automatic recommendation of creation ideas conceptually possible.

In §5.4, Figure 4 shows some questions recommended by FormLM. FormLM is able to recommend questions like "Destination", "Departure Date", "Type of Accommodation" which are highly related to the topic of travelling and can help collect meaningful information for the travel agency. For Options Recommendation, FormLM can accurately identify polar questions and recommend "Yes", "No" as candidate options. Also, since FormLM is continually pre-trained on a large amount of online forms, it has no difficulty recommending options for those frequently asked questions, *e.g.*, "Gender", "Current Educational Qualifications", *etc.*. More interestingly, we notice that FormLM can provide accurate recommendation for questions which are related to their previous contexts. Figure 8 gives two sample outputs by FormLM for Options Recommendation. In the left sample, FormLM gives concrete suggestions which are based on the form title; in the right sample, the recommended locations are all related to school, and they accord well with the domain of this form. We assume that such good performance can be attributed to the effective understanding of form structure and context.

E Human Evaluation

Apart from reporting automatic evaluation results using ROUGE scores, we further conduct human evaluations for Question Recommendation and Options Recommendation. We randomly choose 50 samples from the test sets of the two task and collect the recommended question / options from 5 models (GPT-2, BART$_{\text{BASE}}$, BART, FormLM$_{\text{BASE}}$, FormLM). We use an HTML website (actually an online form service) to collect the manual labels. Human evaluation instructions are shown in Figure 9 and Figure 10. Eight experts familiar with online form software products participate in the experiment. For each sample of a task, we construct a Likert question containing the 5 outputs (randomly shuffled and anonymized) of the models. For each sample, three experts compare the 5 outputs using a rating scale of 1 to 5 (the higher, the better) at the same time to achieve better comparison and annotation consistency across different outputs. So in total, we collect 150 expert ratings for each model on each task.

The evaluation results are shown in Table 6 and Table 7. We can see FormLM and FormLM$_{\text{BASE}}$ outperform all baseline models on both Question and Options Recommendation when manually eval-



RSE Youth and Adult Participant Registration

Event Basics:
Dates: June 10-13, 2018 (June 10 - 4:30pm-8:30pm, June 11-13 - 7:00am-8:00pm, Concert June 13, 8:00pm)
...

Last Name

Your answer

First Name

Your answer

Are you registering with a church?

If you are, please enter church name and city. If you are coming with a friend type in the name of their church. Otherwise just type "None".

Your answer

Gender

○ Male ○ Female

I am a ... ● Choice

Suggested options: Add all | Youth Participant Adult Participant

○ Option 1

+ Add option

Online Bully Report

Choosing to help someone in need is very brave. If you see this happening again, please report it. Together we can stop bullying.

Name of victim(s)

Your answer

Name of Student(s) bullying

Your answer

Select a School

○ Bay High School ○ Bay - Waveland Middle School ...

Date of this incident (as close as possible)

Month, day, year

• • •

Where did the incident happen? ● Choice

Suggested options: Add all | Classroom Hallway Cafeteria Restroom Bus Online

○ Option 1

+ Add option

Figure 8: Sample Outputs by FormLM for Options Recommendation. The suggested options are highlighted in blue.

Rating	5	4	3	2	1	Avg.	≥4	≥3	≤2
GPT-2	16	22	23	20	69	2.31	38	61	89
BART$_{\text{BASE}}$	28	21	12	23	66	2.48	49	61	89
BART	26	23	25	18	58	2.61	49	74	76
FormLM$_{\text{BASE}}$	63	47	13	15	12	3.89	110	123	27
FormLM	72	41	16	9	12	**4.01**	113	129	21

Table 6: Summary of Human Evaluation Ratings for Question Recommendation.

Rating	5	4	3	2	1	Avg.	≥4	≥3	≤2
GPT-2	16	10	6	9	109	1.77	26	32	118
BART$_{\text{BASE}}$	63	28	17	14	28	3.56	91	108	42
BART	68	30	23	9	20	3.78	98	121	29
FormLM$_{\text{BASE}}$	71	35	18	9	17	3.89	106	124	26
FormLM	89	29	14	7	11	**4.19**	118	132	18

Table 7: Summary of Human Evaluation Ratings for Options Recommendation.

uated by the experts, which is in accordance with the automatic evaluation results.

We further conduct Wilcoxon signed-rank test (Woolson, 2007) which is a non-parametric hypothesis test for the matched-pair data to check statistical significance of the comparison between FormLM, FormLM$_{\text{BASE}}$ and their backbone models. At 95% confidence level, when comparing FormLM with BART and comparing FormLM$_{\text{BASE}}$ with BART$_{\text{BASE}}$, both p-values from Wilcoxon test are less than 0.005. These results show that our models have better performance on these two generation tasks than their backbone PLMs which are well-known for conditional generation.

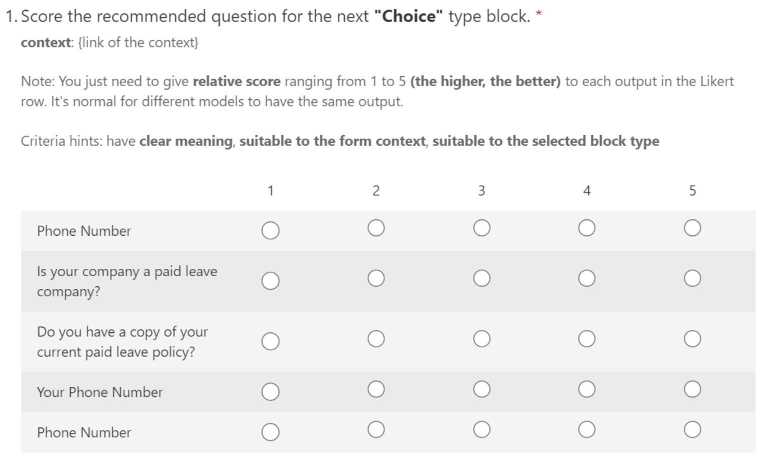
Background

Online forms are widely used to collect data in everyday scenarios and many software products provide services to help users create online forms which consist of multiple blocks. However, for each form question, form designers need to write an informative title, specify its type, and provide other required components. Such a process is time-consuming. Therefore, we want to design a model to **recommend creation ideas and suggestions to online form designers**.

Question Recommendation

Question Recommendation aims at providing users with a recommended question based on the selected block type and the previous context (form title, form description, previous blocks).

In this study, you will evaluate 10 sets of questions recommended by 5 different models. (Model outputs have been randomly shuffled.) The evaluation interface is as follows:



1. Score the recommended question for the next **"Choice"** type block. *

context: {link of the context}

Note: You just need to give **relative score** ranging from 1 to 5 **(the higher, the better)** to each output in the Likert row. It's normal for different models to have the same output.

Criteria hints: have **clear meaning**, **suitable to the form context**, **suitable to the selected block type**

	1	2	3	4	5
Phone Number	○	○	○	○	○
Is your company a paid leave company?	○	○	○	○	○
Do you have a copy of your current paid leave policy?	○	○	○	○	○
Your Phone Number	○	○	○	○	○
Phone Number	○	○	○	○	○

For each sample, you need to

Step 1: Click the link behind “context:” to see the previous context of the form.

Step 2: Check the block type marked in bold black.

Step 3: Score the recommendations. Each row in the Likert table refers to a model output. You can score each output with the relative score ranging from 1 to 5 (higher score indicates better recommended question). **Note that your score should consider three parts**:

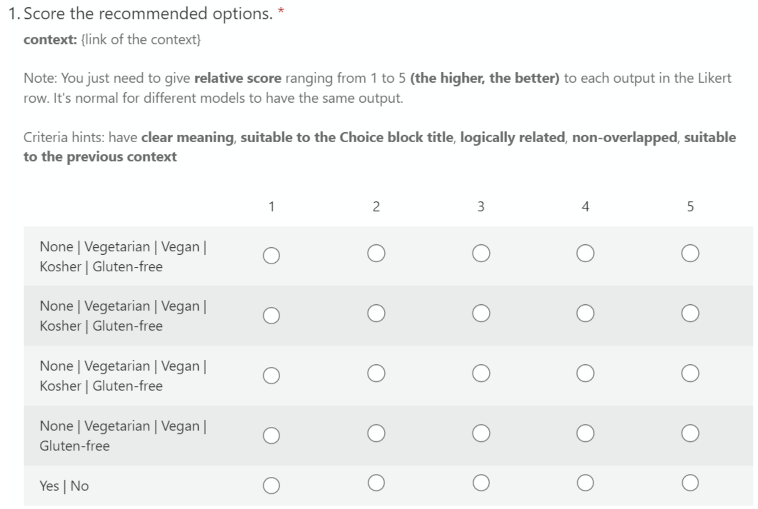
- Whether the question has clear meaning.
- Whether the question is suitable to the form context (relevant to the form title, non-overlap with previous questions, logically coherent with previous questions, etc.).
- Whether the question suits the selected block type.

Figure 9: Human Evaluation Instructions. (Page 1 / 2)

Options Recommendation

Choice blocks are frequently used in online forms. When creating a Choice block, one should additionally provide a set of options. Options Recommendation aims recommending a set of options to users based on the current block title and all the previous context (form title, form description, previous blocks).

In this study, you will evaluate 10 sets of questions recommended by 5 different models. (Model outputs have been randomly shuffled.) The evaluation interface is as follows:



1. Score the recommended options. *

context: {link of the context}

Note: You just need to give **relative score** ranging from 1 to 5 **(the higher, the better)** to each output in the Likert row. It's normal for different models to have the same output.

Criteria hints: have **clear meaning**, **suitable to the Choice block title**, **logically related**, **non-overlapped**, **suitable to the previous context**

	1	2	3	4	5
None | Vegetarian | Vegan | Kosher | Gluten-free	○	○	○	○	○
None | Vegetarian | Vegan | Kosher | Gluten-free	○	○	○	○	○
None | Vegetarian | Vegan | Kosher | Gluten-free	○	○	○	○	○
None | Vegetarian | Vegan | Gluten-free	○	○	○	○	○
Yes | No	○	○	○	○	○

For each sample, you need to

Step 1: Click the link behind "context:" to see the previous context of the form and the choice block title that models will make recommendations for.

Step 2: Score the recommendations. Each row in the Likert table refers to a model output. Note that we expect models to recommend a set of options, and we concatenate the options with a vertical bar "|". You can score each output with the relative score ranging from 1 to 5 (higher score indicates better recommended options). **Note that your score should consider three parts**:

- Whether each option has clear meaning and whether it is a suitable answer to the Choice block title.
- Whether this set of options are logically related to each other and non-overlapped.
- Whether this set of options are reasonable when considering the previous form context.

Figure 10: Human Evaluation Instructions. (Page 2 / 2)