TOWARD CONTROLLABLE TEXT CONTENT MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Controlled generation of text is of high practical use. Recent efforts have made impressive progress in generating or editing sentences with given textual attributes (e.g., sentiment). This work studies a new practical setting of *text content manipulation*. Given a structured record, such as *(PLAYER: Lebron, POINTS: 20, ASSISTS: 10)*, and a reference sentence, such as *Kobe easily dropped 30 points*, we aim to generate a sentence that *accurately* describes the full content in the record, with the *same* writing style (e.g., wording, transitions) of the reference. The problem combines the characteristics of data-to-text generation and style transfer, and is challenging to minimally yet effectively manipulate the text (by rewriting/adding/deleting text portions) to ensure fidelity to the structured content. We derive two datasets from the data-to-text task as our testbed, and develop a neural method with weakly supervised competing objectives and explicit content coverage constraints. Automatic and human evaluations show superiority of our approach over competitive methods including a strong rule-based baseline and prior approaches designed for style transfer.

1 INTRODUCTION

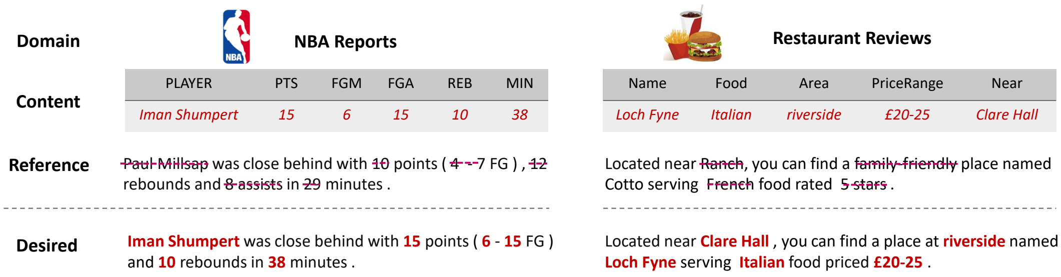
Though significant progress of natural language generation has been made in recent years (Sutskever et al., 2014; Vaswani et al., 2017), which is further boosted by the large scale model pre-training technique (Radford et al., 2018), generating natural language with controllability is still under investigated. Controllability with existing models is mainly achieved by conditioning the generation on given inputs, which does not guarantee controll In this paper, we investigate controlled generation text generation problem in the context of language generation of structured information. Generating natural language text to describe structured content, such as a database record or a table, is of ubiquitous use in real-life applications including data report generation (Wiseman et al., 2017), article writing (Lebret et al., 2016; Kiddon et al., 2016), dialog systems (Wen et al., 2015; Yang et al., 2016), and many others. Recent efforts have developed many techniques to improve *fidelity* to the source content, such as new powerful neural architectures (Gu et al., 2016; See et al., 2017), hybrid generation and retrieval (Hashimoto et al., 2018; Weston et al., 2018), and so forth, most of which are applied in supervised context.

Language is rich with variation–given a data record, there are diverse possible ways of saying the same content[1], with different word choices, expressions, transitions, tone, etc[2]. Previous data-to-text work has largely focused only on content fidelity, while ignoring and lacking control over the rich stylistic properties of language. It can be practically useful to generate text that is not only describing the conditioning content, but also following a designated writing style.

In this work, we study the new yet practical problem in which we aim to express given content with a sentence and mimic the writing style of a reference sentence (Fig 1). More specifically, we are given a structured data record containing the content to describe, along with a sentence about a similar but different matter. Our goal is to generate a new sentence that precisely depicts all content in the record, while at the same time using as much of the writing style of reference sentence as possible. As above, the problem differs critically from the supervised data-to-text (Wiseman et al.,

[1] `https://en.wikipedia.org/wiki/Variation_(linguistics)`

[2] We refer to these characteristics that are independent of the desired content as *writing style*.



PLAYER	PTS	FGM	FGA	REB	MIN
Iman Shumpert	*15*	*6*	*15*	*10*	*38*

~~Paul Millsap~~ was close behind with ~~10~~ points (~~4 -~~ 7 FG) , ~~12~~ rebounds and ~~8 assists~~ in ~~29~~ minutes .

Iman Shumpert was close behind with **15** points (**6** - **15** FG) and **10** rebounds in **38** minutes .

Name	Food	Area	PriceRange	Near
Loch Fyne	*Italian*	*riverside*	*£20-25*	*Clare Hall*

Located near ~~Ranch~~, you can find a ~~family-friendly~~ place named Cotto serving ~~French~~ food rated ~~5 stars~~ .

Located near **Clare Hall** , you can find a place at **riverside** named **Loch Fyne** serving **Italian** food priced **£20-25** .

Figure 1: An example input (content record and reference sentence) of text content manipulation andits desired output. Text portions that fulfill the writing style are highlighted in right.

2017) or retrieval-and-rewriting work (Hashimoto et al., 2018; Weston et al., 2018) as we have imposed an additional goal of preserving the reference text style. The resulting problem is typically *unsupervised* due to lack of parallel data.

The problem also differs in important ways from the emerging task of *text style transfer* (Hu et al., 2017; Shen et al., 2017) which assumes an existing sentence of certain content, and modifies single or multiple textual attributes of the sentence (e.g., transferring negative sentiment to positive) without changing the content. Our task, on the contrary, assumes abstract style is encoded in a reference sentence and attempts to modify its concrete content to express new information from the structured record. The different setting can lead to different application scenarios in practice, and pose unique technical challenges. In particular, though the most recent style transfer research (Subramanian et al., 2019; Logeswaran et al., 2018) has controlled multiple categorical attributes which are largely independent or loosely correlated to each other, a content record in our task, in comparison, can contain *varying* number of entries which are of different types (e.g., player, points, defensive/offensive rebounds, etc), having many possible values (e.g., hundreds of players), and are structurally coupled (e.g., 32 points by Lebron). A model must understand the content structure, and minimally yet sufficiently manipulate the reference sentence by rewriting, adding, or deleting text portions, with necessary polishing for grammatical correctness and fluency. We name the problem *text content manipulation*. Our empirical studies show the most recent models designed for style transfer fail to perform well in the task.

In this paper, we first develop a large unsupervised dataset as a testbed of the new task. The dataset is derived from an NBA game report corpus (Wiseman et al., 2017). In each data instance, besides a content record and a reference sentence as the problem inputs, we also collect side information useful for unsupervised learning. Specifically, each instance has an auxiliary sentence that was originally written by human reporters to describe the content record without seeing (and thus stylistically irrelevant to) the reference sentence. We also provide the structured record of the reference sentence. The side information can provide valuable clues for models to understand the content structure and text semantics at training time. We do not rely on the side information at test time.

We then propose a neural method to tackle the problem. With a hybrid attention and copy mechanism, the model effectively encodes the reference and faithfully copies content from the record. The model is learned with two competing objectives of reconstructing the auxiliary sentence (for content fidelity) and the reference sentence (for style preservation). We further improve the model with an explicit content coverage constraint which encourages to precisely and fully convey the structured content.

For the empirical study, we devise automatic metrics to measure content fidelity and style preservation, respectively. We also perform human evaluations to compare different approaches. Results demonstrate the proposed method significantly improves over others, including a strong rule-based baseline and the recent style transfer models.

2 Related Work

Generating text conditioning on structured input has been widely studied in recent work (Wen et al., 2015; Lebret et al., 2016; Yang et al., 2016; Wiseman et al., 2017, etc). Those methods are based on neural sequence to sequence models and trained with supervised data. This line of work has focused primarily on generating more accurate description of the given data, while it has not studied the problem of controlling the writing style of outputs. Our task takes a step forward to *simultaneously* describing desired content and controlling stylistic properties. Furthermore, our task is challenging due to its unsupervised setting in practice.

Beyond generating text from scratch, there is another line of work that first retrieves a similar sentence and then rewrites it to express desired information (Hashimoto et al., 2018; Weston et al., 2018; Li et al., 2018; Guu et al., 2018). For example, Weston et al. (2018) used the framework to generate response in dialogues, while Hashimoto et al. (2018) studied programming code generation. The goal of the work is to manifest useful information from neighbors, usually in a supervised context, without aiming at controlling writing characteristics, and thus has fundamentally different assumptions to ours.

Recently, there has been growing interest in text style transfer, in which many techniques for controlled text generation are developed (Hu et al., 2017; Shen et al., 2017; Yang et al., 2018; Prabhumoye et al., 2018; Tian et al., 2018; Subramanian et al., 2019; Logeswaran et al., 2018). The main idea underlying those models is to learn disentangled representations of text so as modify textual attributes or style of interest. Those papers used different objectives to encourage learning disentangled representations. Hu et al. (2017) used pre-trained classifiers as the supervision. Shen et al. (2017) used a GAN-based approach in which binary classifiers were used as discriminators. Yang et al. (2018) proposed to use more structured discriminators such as language models to provide better supervision to the generator. Prabhumoye et al. (2018); Subramanian et al. (2019) further augmented prior work using back-translation technique to incorporate cycle-consistency loss. Both (Subramanian et al., 2019) and (Logeswaran et al., 2018) generalized the task to controlling multiple categorical attributes at the same time. Our work differs from those in that we assume an existing sentence to provide the source of style and a structured record as the source of content. The input content record in our task is also more structured than the style attributes which are typically loosely connected and of a pre-fixed number. The resulting content manipulation setting poses unique challenges in controlling, as discussed more in the empirical study.

3 Task Definition: Text Content Manipulation

We first formally define the problem of unsupervised text content manipulation, and establish the notations. We then illustrate the generality of the task via comparing with two exiting tasks.

Without loss of generality, consider a content record $\boldsymbol{x} = \{\boldsymbol{x}_i\}_{i=1}^{L_x}$, where each element $\boldsymbol{x}_i$ is a data tuple which typically includes a data type (e.g., points), a value (e.g., 32), and other information (such as the associated player, e.g., Lebron_James). L_x is the number of tuples in record $\boldsymbol{x}$, which can vary across different records. We are also given a reference sentence $\boldsymbol{y}'$ which is assumed to describe content that has a similar but not exact the same structure with that of the record $\boldsymbol{x}$. For example, in Fig 1, both the content record and the reference sentence involve two players, respectively, but the number of associated data tuples as well as the types are different (e.g., *Lebron_James* in the record has 3 box-score entries, while *Jrue_Holiday* in the reference has only 2).

We may also have access to other side information at training time. For example, in the dataset developed below, each content record $\boldsymbol{x}$ is associated with an auxiliary sentence $\boldsymbol{y}$ that was originally written to describe $\boldsymbol{x}$ without following the reference $\boldsymbol{y}'$. Each reference sentence $\boldsymbol{y}'$ also has its corresponding record $\boldsymbol{x}'$ containing the content information. The side information can provide valuable clues for models to understand the content structure and text semantics at training time. For example, the auxiliary sentence provides a hint on how the desired content can be presented in natural language, though it is stylistically irrelevant to the reference sentence. Note that, at test time, a solution to the task should only rely on the inputs $(\boldsymbol{x}, \boldsymbol{y}')$ without using the side information.

The goal of the task is to generate a new realistic sentence $\hat{\boldsymbol{y}}$ that achieves **(1) content fidelity** by accurately describing the full content in $\boldsymbol{x}$, and at the same time **(2) style preservation** by retaining

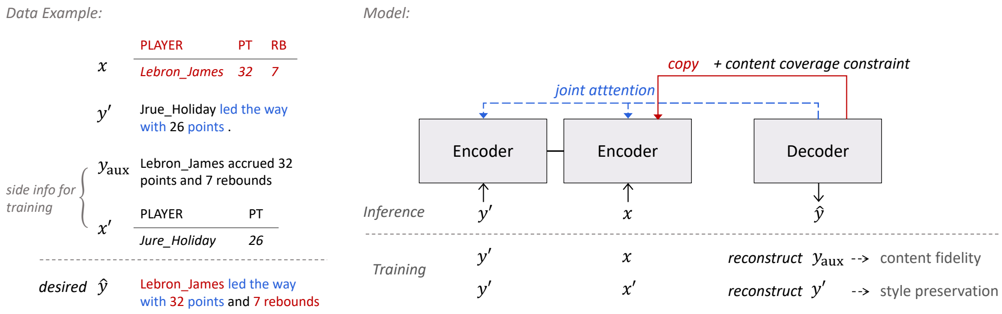

Data Example:
PLAYER PT RB
x
Lebron_James 32 7
y'
Jrue_Holiday led the way with 26 points .
y_{aux}
Lebron_James accrued 32 points and 7 rebounds
side info for training
PLAYER PT
x'
Jure_Holiday 26
desired $\hat{y}$
Lebron_James led the way with 32 points and 7 rebounds
Model:
copy
+ content coverage constraint
joint atttention
Encoder
Encoder
Decoder
Inference
y'
x
$\hat{y}$
Training
y'
x
reconstruct y_{aux} --> content fidelity
y'
x'
reconstruct y' --> style preservation

Figure 2: A (simplified) data example (left) and the model overview (right).

as much of the writing style and characteristics of reference $\boldsymbol{y}'$ as possible. The task is unsupervised as there is no ground-truth sentence for training.

4 Proposed Model

We next develop methods to tackle the problem. As shown in the empirical study (section 6), a simple rule-based method that matches $\boldsymbol{x}$ with $(\boldsymbol{x}', \boldsymbol{y}')$ and performs text replacement would fail in terms of content fidelity due to the different structures between $\boldsymbol{x}$ and $\boldsymbol{x}'$. Previous approaches for (multi-attribute) style transfer do not apply well either, because of the different underlying task assumptions and the rich content structures of records with varying lengths.

In the following, we present a new neural approach that addresses the challenges of text content manipulation. We first describe the model architecture, then develop unsupervised learning objectives, and finally add a content coverage constraint to improve learning. Figure 2 provides an illustration of the proposed approach.

Let $p_\theta(\hat{\boldsymbol{y}}|\boldsymbol{x}, \boldsymbol{y}')$ denote the model that takes in a record $\boldsymbol{x}$ and a reference sentence $\boldsymbol{y}'$, and generates an output sentence $\hat{\boldsymbol{y}}$. Here $\boldsymbol{\theta}$ is the model parameter.

4.1 Hybrid Attention-Copy Mechanism

As shown in the figure, the architecture of our neural model consists of two encoders and one decoder. The first encoder is used to extract representation of the reference sentence $\boldsymbol{y}'$. The second encoder is applied to content records. Specifically, for each data tuple in a record, we first concatenate the embedding vectors of all fields in the tuple, and feed the combined embedding to the encoder. There is no need to specify a particular order of the tuples as their fields have specified the associated player or team.

The decoder is to generate the output sentence, with a hybrid attention-copy mechanism at each decoding step. In particular, the decoder applies a joint attention (Luong et al., 2015) over both $\boldsymbol{y}'$ and $\boldsymbol{x}$, and applies a copy mechanism (Gu et al., 2016) only on the record $\boldsymbol{x}$. More specifically, at each step t, the decoder first attends jointly to the hidden states of both encoders and obtains a decoding hidden state $\boldsymbol{h}_t$. The decoder then computes the output distribution over words with

$$\boldsymbol{p}_{out}^{(t)} = g_t \cdot \boldsymbol{p}_V^{(t)} + (1 - g_t) \cdot \boldsymbol{p}_{\boldsymbol{x}}^{(t)}, \tag{1}$$

where g_t is the probability of generating a token from the vocabulary; $\boldsymbol{p}_V^{(t)}$ is the generation distribution over the whole vocabulary; while $\boldsymbol{p}_{\boldsymbol{x}}^{(t)}$ is the copy distribution over the data values in the content record. All the quantities are computed based on the post-attention hidden state $\boldsymbol{h}_t$.

4.2 Competing Learning Objectives

As defined in section 3, the task has two simultaneous goals, namely data fidelity and style preservation. The two goals are in a sense competitive with each other (e.g., describing the new designated content would usually change the expressions in reference sentence to some extent). We base our unsupervised learning on this competitive relation.

We make use of the side information $(\boldsymbol{y}_, \boldsymbol{x}')$ during training. Specifically, as the auxiliary sentence $\boldsymbol{y}$ was originally written by human to describe the content $\boldsymbol{x}$ and thus can be seen to have the maximum data fidelity, we devise the first objective that reconstructs $\boldsymbol{y}$ given $(\boldsymbol{x}, \boldsymbol{y}')$:

$$\mathcal{L}_{content}(\boldsymbol{\theta}) = \log p_\theta(\boldsymbol{y}_{|}\boldsymbol{x}, \boldsymbol{y}'). \tag{2}$$

We call it the content fidelity objective.

To fulfill the second goal of preserving the style of reference $\boldsymbol{y}'$, we want to encourage the model to generate sentences in a similar form of $\boldsymbol{y}'$. We further notice that, if we feed the model with the reference sentence $\boldsymbol{y}'$ and its corresponding record $\boldsymbol{x}'$ (instead of $\boldsymbol{x}$), the ground truth output of the case is indeed $\boldsymbol{y}'$ itself (as $\boldsymbol{y}'$ describes content $\boldsymbol{x}'$, and is of course in the same style of itself). We thus can specify the second objective that reconstructs $\boldsymbol{y}'$ given $(\boldsymbol{x}', \boldsymbol{y}')$:

$$\mathcal{L}_{style}(\boldsymbol{\theta}) = \log p_\theta(\boldsymbol{y}'|\boldsymbol{x}', \boldsymbol{y}'). \tag{3}$$

We call it the style preservation objective. The objective essentially treats the reference sentence encoder and the decoder together as an auto-encoding module, and effectively drives the model to absorb the characteristics of reference sentence and apply to the generated one.

The above two objectives are coupled together and train the model to achieve the desired goals:

$$\mathcal{L}_{joint}(\boldsymbol{\theta}) = \lambda\mathcal{L}_{content}(\boldsymbol{\theta}) + (1-\lambda)\mathcal{L}_{style}(\boldsymbol{\theta}), \tag{4}$$

where λ is the balancing parameter.

4.3 Content Coverage Constraint

As shown in the empirical study (section 6), the above learning can yield reasonably good performance, but sometimes can still fall short of accurately expressing the desired content. We thus devise an additional learning constraint based on the nature of content description—each data tuple in the content record should usually be mentioned *exactly once* in the generated sentence.

The copy mechanism over content record $\boldsymbol{x}$ enables a simple yet effective way to encourage the behavior. Intuitively, we want each tuple to be copied once and only once on average. We thus minimize the following L2 constraint that drives the aggregated copy probability of each data tuple to be 1:

$$\mathcal{C}(\boldsymbol{\theta}) = \left\|\sum\nolimits_t \boldsymbol{p}_{\boldsymbol{x}}^{(t)} - \mathbf{1}\right\|^2, \tag{5}$$

where $\boldsymbol{p}_{\boldsymbol{x}}^{(t)}$, as defined in Eq.equation 1, denotes the copy distribution over all data tuples at decoding step t; and $\mathbf{1}$ is a vector with all ones. It is still possible that tokens of the content values "leak" from the generation distribution $\boldsymbol{p}_V^{(t)}$ in Eq.equation 1. We disable the leakage by masking out relevant words (particularly numbers) for each instance from the vocabulary.

We note that prior work in other context, especially machine translation, has also explored the idea of *coverage* through either architecture augmentation (Tu et al., 2016) or inference penalty (Wu et al., 2016). We tried these techniques but did not obtain noticeable improvement. As shown in the experiments, the proposed explicit coverage constraint over copy mechanism leads to significant performance gains.

The full training objective of the proposed model with the constraint is thus written as:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{joint}(\boldsymbol{\theta}) - \eta \cdot \mathcal{C}(\boldsymbol{\theta}), \tag{6}$$

where η is the weight of the constraint.

5 Dataset

We now present two datasets developed for the task[3]. The first dataset is derived from a recent large table-to-document corpus (Wiseman et al., 2017) which consists of box-score tables of NBA basketball games and associated documents as game reports. The corpus is originally used for studying

[3] The dataset is released at https://github.com/ZhitingHu/text_content_manipulation

	NBA Reports			**Restaurant Reviews**		
	Train	**Dev**	**Test**	**Train**	**Dev**	**Test**
#Instances	31,444	6,765	6,930	29,486	6,299	6,273
#Tokens	7.88M	1.69M	1.75M	0.54M	0.12M	0.12M
Avg Sentence Length	25.07	25.10	25.32	18.36	18.34	18.35
#Data Types	34	34	34	8	8	8
Avg Record Length	4.32	4.31	4.35	5.38	5.38	5.35

Table 1: The data statistics for two different datasets, which both derived from the exiting data-to-text datasets.

supervised game report generation which has attracted increasing research interest (Nie et al., 2018; Wiseman et al., 2017). The second dataset is derived from the E2E Generation dataset (Dušek et al., 2019), which is a crowd-sourced dataset in the restaurant domain.

NBA News Reports To obtain our NBA data, we first split each game report into individual sentences, and, for each sentence, find its corresponding data in the box-score table as the content record. A record can contain a varying number of tuples, with each tuple containing three fields, namely a data type, a value, and an associated player or team, e.g., *(team_points, 106, Lakers)*. As the original corpus is already largely clean, we found some simple rules are sufficient to obtain high-quality results in this step. Please see the supplementary materials for more details. Each of the resulting record-sentence pairs is treated as a pair of $(\boldsymbol{x}, \boldsymbol{y}_))$, namely (content record, auxiliary sentence). The next step is to find a suitable reference sentence $\boldsymbol{y}'$ for each content record $\boldsymbol{x}$. As defined above, the reference sentence should cover similar but not the same content as in record $\boldsymbol{x}$. We achieve this by retrieving from the data another record-sentence pair using $\boldsymbol{x}$, where the retrieved record is designated to have a slightly different structure than that of $\boldsymbol{x}$ by having less or more tuples and different data types. More details of the retrieval method are deferred to supplements. The retrieved record-sentence pair thus plays the role of $(\boldsymbol{x}', \boldsymbol{y}')$ and is paired with $(\boldsymbol{x}, \boldsymbol{y}_))$ to form an instance.

Restaurant Reviews riginal E2E dataset consists of the review records, e.g.(*eatType, coffee shop, Bibimbap House*) and natural language descriptions of restaurants. It enables us to explicitly construct the record-sentence pairs $(\boldsymbol{x}, \boldsymbol{y}_))$. We construct this data set in a similar way to the NBA new reports data set. In contrast, to make this task more challenging, we adopt a new retrieval method in constructing $\boldsymbol{y}'$, that is, we constraint the corresponding $\boldsymbol{x}'$ to have at at least 2 different types of entries compared with $\boldsymbol{x}$.

6 Experiments

We conduct both automatic and human evaluations to assess the model performance. For automatic evaluation, we use two metrics to measure content fidelity and style preservation, respectively. Results show our model balances well between the two goals, and outperforms a variety of comparison methods. All code will be released soon.

6.1 Experimental Setup

Comparison Approaches
We compare with a diverse set of approaches:

- **AttnCopy-S2S.** We first evaluate a base sequence-to-sequence (Sutskever et al., 2014) model with the above attention-copy mechanism, which takes in record $\boldsymbol{x}$ and generates its descriptive sentence $\boldsymbol{y}$. The evaluation provides a sense of the difficulty in describing desired content.
- **Rule-based Method.** A straightforward way for text content manipulation is to match between $\boldsymbol{x}$, $\boldsymbol{x}'$ and $\boldsymbol{y}'$ with certain rules, and replace corresponding portions in $\boldsymbol{y}'$ with those in $\boldsymbol{x}$. Specifically, we first build a mapping between the tuples of $\boldsymbol{x}$ and $\boldsymbol{x}'$ through their data types, and a mapping between $\boldsymbol{x}'$ and $\boldsymbol{y}'$ through data values, types and indicative tokens (e.g., "12 points" in $\boldsymbol{y}'$ indicates 12 is of type player points or team_points). The two mappings connect $\boldsymbol{x}$ and $\boldsymbol{y}'$, enabling us to swap appropriate text in $\boldsymbol{y}'$ to express content $\boldsymbol{x}$.

In theory, rule-based method sets the best possible style preservation performance, as it only replaces content related tokens (particularly numbers) without modifying other parts of the reference sentence. The output, however, tends to miss or contain extra content compared to the content record of interest.

- **Multi-Attribute Style Transfer (MAST)** (Subramanian et al., 2019). We compare with the most recent style transfer approach that models multiple attributes. To apply to our setting, we treat content record $\boldsymbol{x}$ as the attributes. The method is based on back-translation (Sennrich et al., 2015) that first generates a target sentence $\hat{\boldsymbol{y}}$ conditioning on $(\boldsymbol{x}, \boldsymbol{y}')$, and then treat it as the reference to reconstruct $\boldsymbol{y}'$ conditioning on $(\boldsymbol{x}', \hat{\boldsymbol{y}})$. Auxiliary sentence $\boldsymbol{y}$ is used in an extra auto-encoding loss.
- **Adversarial Style Transfer (AdvST)** (Logeswaran et al., 2018). As another latest style transfer approach capable of handling more than one attributes, the model also mixes back-translation with auto-encoding as the above method, and additionally uses adversarial training to disentangle content and style representations.
- **Ours LSTM w/o Coverage.** For ablation study, we compare with our LSTM model variant that omits the content coverage constraint. That is, the model is trained by maximizing only Eq.equation 4.
- **Ours Transformer w/o Copy.** We empirically found that disenabling the self-attention to $\boldsymbol{y}'$ only at the final decoding block can achieve the similiar copying effects. The results are resonable , since this operation enables it attending to $\boldsymbol{x}$ to the greatest extent. Hence, we compare with this kind of standard transformer model.
- **Ours Transformer w/o Coverage.** For ablation study, we compare with our transformer model variant that omits the content coverage constraint . That is, the model is trained by maximizing only Eq.equation 4.

Model Configurations
We employ experiments on both LSTM and Transformer. For LSTM, we used are single-layer encoder-decoder architecture with the Luong attention (Luong et al., 2015). For Transformer, we set the number of blocks in encoders and decoders to be 3. For both models, we employ the copy mechanism Vinyals et al. (2015) to improve performance. Both the embedding dimension and hidden dimension are set to 384. During training, we first set $(\lambda = 0, \eta = 0)$ to pre-train the model to convergence so that the model captures the full characteristics of the reference sentence. Then we switch to set $(\lambda = 0.2, \eta = 1.0)$ for full training. We apply Adam optimization (Kingma & Ba, 2014) with an initial learning rate of 0.001 and gradient norm clipping of 15. For inference we use beam search with beam-width 5. The maximum decoding length is set to 50.

6.2 Automatic Evaluation

As no ground truth annotations are available, we first set up automatic metrics for quantitatively measuring the key aspects of model performance.

Metrics
We propose metrics to evaluate the two primary goals of the task, namely content fidelity and style preservation. A desired solution should balance and excel on both metrics.

- **Content fidelity.** For the NBA dataset, we use an information extraction (IE) approach to measure content fidelity following the data-to-text task (Wiseman et al., 2017) . That is, given a generated sentence $\hat{\boldsymbol{y}}$ and the conditioning content record $\boldsymbol{x}$, we extract data tuples from $\hat{\boldsymbol{y}}$ with an IE tool, and compute the precision and recall against $\boldsymbol{x}$. We use the IE model provided in (Wiseman et al., 2017) , which achieves around 81% precision and 86% recall on the test set. For the Restaurant dataset, we train a binary transformer classifier to evaluate whether $\hat{\boldsymbol{y}}$ manipulate the content successfully or not. The classifier achieves 94% accuracy on the test set. Specifically, we define the content fidelity score as the accuracy of overall new content records of $\boldsymbol{x}$ that are expressed by $\hat{\boldsymbol{y}}$. The content residual score is the percentage of old content records that should be deleted in $\hat{\boldsymbol{y}}$. The desired model should express the more new content records and less old ones. Hence, lower residual score represent the stronger ability of manipulating the old content.

		NBA Reports			**Restaurant Reviews**		
		Content		Style	Content		Style
	Model	**Precision%**	**Recall%**	**BLEU**	**Fidelity Score%**	**Residual Score%**	**BLEU**
1	AttnCopy-S2S	81.62±3.25	75.65±7.42	45.5±0.71	78.88±2.08	0.29±0.06	13.95±0.52
	Rule-based	56.69	71.34	100	61.23	33.8	100
2	MAST	23.06±3.90	27.37±3.88	**95.43±2.71**	36.28±0.25	62.94±0.16	**91.76±0.28**
	AdvST	67.37±0.66	66.79±1.43	64.67±4.81	51.64±4.45	34.23±4.44	76.02±5.27
3	**Our results:**						
	LSTM-based.	68.74±3.07	69.35±3.30	79.88±2.44	60.83±1.29	18.55±1.10	78.91±1.05
	+ Cover	**69.54±1.16**	73.27±1.18	80.66±1.89	**65.02±4.16**	**17.47±0.70**	82.92±3.18
4	Transformer-based.	62.58±2.88	70.22±3.58	81.75±2.32	60.03±2.16	25.35±2.69	77.81±3.83
	+ Copy.	65.76±2.45	73.61±0.08	81.10±2.87	61.96±0.23	25.31±1.17	79.15±1.49
	+ Copy + Cover	67.74±0.79	**74.35±1.22**	81.97±2.87	61.84±1.31	21.86±2.73	80.29±0.35

Table 2: Model performance under automatic evaluation. Results are averaged over 3 runs ± one standard deviation. Except for the content score 2, the higher the score is, the better performance is. Models in the first block (AttnCopy Seq2seq and Rule-based) represent two baselines for reference performance. We have highlighted the best results in blocks 2, 3 and 4 under different metrics. Our two models with LSTM or Transformer achieve significant higher precision and recall respectively compared to both rule-based and style transfer methods, and reaches a high BLEU score in style preservation.

- **Style preservation.** A generated sentence is desired to retain stylistic properties, such as word choice and expressions, of the input reference sentence. Inspired by the text style transfer literature (Yang et al., 2018; Subramanian et al., 2019), we measure the BLEU score between generated and reference sentences. To reduce the influence of new content, we first mask in both sentences all obvious content tokens, including player/team names and numbers, by replacing them with a special token `<M>`, and then compute the BLEU score. In this way, the above rule-based method has a maximum BLEU score of 100, which is consistent with our intuition above.

Results

We now compare the performance of different methods in terms of the above metrics. Table 2 shows the quantitative results obtained on NBA and Restaurant datasets.

The first block shows the two baseline models providing reference performance. The AttnCopy-S2S model only concerns about content fidelity, and achieves a high content precision score (but a low recall). However, its style BLEU is particularly low, which verifies the rich variation in language and that direct supervised learning is incapable of controlling the variation. As discussed above, the rule-based method can reach the maximum BLEU (100) after masking out content tokens while its content precision score is unsatisfactory. The two style transfer methods (MAST and AdvST) fail the expectation, as their content fidelity performance is greatly inferior or merely comparable to the rule-based method. This is partially because these models are built on a different task assumption (i.e., modifying independent textual attributes) and cannot manipulate content well.

For the NBA dataset, our proposed LSTM/Transformer models achieve better content precision/recall, substantially improving over other methods (e.g., with a 12-point precision boost in comparison with the rule-based baseline) except for AttnCopy-S2S which has failed in style control. Our methods also manage to preserve a high BLEU score of over 80. The superior performance of the full model on both datasets compared to the variant Ours-w/o-Coverage demonstrates the usefulness of the content coverage constraint (Eq.5). By explicitly encouraging the model to mention each of the data tuples exactly once—a common pattern of human-written descriptions—the model achieves higher content fidelity with less style-preservation ability "sacrificed". We see similar trend for the Restaurant dataset, our proposed LSTM/Transformer based models achieve better fidelity and residual score. One of the reason why our proposed Transformer-based model is worse than LSTM-based model may be owing to the fact that restaurant dataset has shorter sentence length and this may limit the power of Transformer in capturing the long semantic dependency.

	NBA Reports			Restaurant Reviews		
Model	Content Fidelity	Style Preserv.	Fluency	Content Fidelity	Style Preserv.	Fluency
Rule-based	2.79	**5.00**	**4.96**	3.36	**5.00**	**4.70**
AdvST	2.88	4.00	4.09	3.56	4.24	4.02
Ours:						
LSTM w/o Cover.	3.43	4.13	4.59	3.91	4.38	4.48
LSTM	**3.88**	4.53	4.45	**4.08**	4.73	4.34
	Ours Better	**No Prefer.**	**Ours Worse**	**Ours Better**	**No Prefer.**	**Ours Worse**
Rule-based	**67.5%**	17.5%	15.0%	**64.1%**	18.6%	17.3%
AdvST	**68.8%**	17.5%	13.8%	**70.4%**	14.3%	15.2%
LSTM w/o Cover.	**51.3%**	32.5%	16.3%	**52.0%**	26.7%	21.3%

Table 3: Human Evaluation Results. **Top:** Humans are asked to score the model outputs in terms of content fidelity, style preservation, and fluecny, respectively, from 1 (strongly bad) to 5 (strongly good). As expected, the rule-based method reaches the maximum possible scores in terms of style preservation and fluency, but a much lower score in terms of content fidelity. Our LSTM-based model is more balanced across all aspects, and performs significantly better in accurately describing desired content. **Bottom:** Humans are asked to rank a pair of generated sentences in which one is from our model and the other from the comparison method. Our model wins on more than 50% instances compared to each of other models on two datasets.

6.3 Human Evaluation

We also carried out human evaluation for a more thorough and accurate comparison. Following the experimental settings in prior work (Subramanian et al., 2019; Logeswaran et al., 2018; Shen et al., 2017), we undertook two types of human studies: (1) We asked three human turkers to directly score generated sentences in three aspects, namely content fidelity, style preservation, and sentence fluency. Each score is from 1 (strongly bad) to 5 (strongly good); (2) We present to annotators a pair of generated sentences, one from our model and the other from a comparison method, then ask the annotators to rank the two sentences by considering all the criteria. Annotators can also choose "no preference" if the sentences are equally good or bad. For each study, we evaluate on 80 test instances, and compare our model with the rule-based method, AdvST style transfer model (which has shown better performance on the task than the other style transfer model MAST), and the model variant without coverage constraint.

Table 3 shows the human evaluation results. From the top block of the table, as expected and discussed above, the rule-based method sets the records of style preservation and fluency scores, as it only conducts lightweight token replacement on reference sentences. However, its content fidelity score is very low. In contrast, our model achieves a reasonably high content score of 3.88 and 4.08, which is much higher than those of other methods. The model is also more balanced across the three criteria, achieving reasonably high scores in both style preservation and language fluency. The fluency of the full model is slightly inferior to the variant without coverage constraint, which is not unexpected since the full model has modified more portions of reference sentence in order to better describe the desired content, which would tend to introduce more language mistakes as well.

The bottom block of Table 3 shows the results of ranking sentence pairs. We can see that our model consistently outperforms the comparison methods with over 50% wins.

6.4 Qualitative Study

Table 4 shows example outputs on two test cases given content record $\boldsymbol{x}$ and reference sentence $\boldsymbol{y}'$. We can see that, in general, the proposed full model can manipulate the reference sentence more accurately to express the new content. For example, in the first case, the rule-based method was confused between the winning and losing teams, due to its incapacity of understanding the semantics of text such as "held off". The style transfer model AdvST failed to comprehend the content record well and generated irrelevant data "100 - 100". The simplified variant without explicit coverage

Content $\boldsymbol{x}$	**TEAM**	**WINS**	**LOSSES**	**TEAM-PTS**	**TEAM**	**WINS**	**LOSSES**	**TEAM-PTS**	**TEAM**
	Bulls	29	16	102	Mavericks	30	14	98	Dallas
Reference $\boldsymbol{y}'$	The Pistons (22 - 33) held off the Bulls (34 - 21) 100 - 91 in Detroit on Friday night .								
Rule-based	The Mavericks (30 - 14) held off the Bulls (29 - 16) 98 - 102 in Dallas on Friday night .								
AdvST	The Bulls (29 - 16) held off the Mavericks (30 - 14) 100 - 100 in Dallas on Friday night .								
Ours w/o Cover.	The Bulls (29 - 16) held off the Bulls (29 - 16) 102 - 98 in Dallas on Friday night .								
Ours	The Bulls (29 - 16) held off the Mavericks (30 - 14) 102 - 98 in Dallas on Friday night .								

Table 4: Example outputs. Text of erroneous content is highlighted in red, where [...] indicates desired content is missing. Text portions in the reference sentences and the generated sentences by our model that fulfill the stylistic characteristics are highlighted in blue. Please see the text for more details.

constraint copied the content of Bulls twice. In contrast, the full model successfully generates the desired sentence.

7 Discussions

We have proposed a new and practical task of text content manipulation which aims to generate a sentence that describes desired content from a structured record (content fidelity) and meanwhile follows the writing style of a reference sentence (style preservation). To study the unsupervised problem, we derived a new dataset, and developed a method with competing learning objectives and an explicit coverage constraint. For empirical study, we devised two automatic metrics to measure different aspects of model performance. Both automatic and human evaluations showed superiority of the proposed approach.

There are multiple directions to further improve and generalize the work, including enhancing text generation quality in general with more sophisticated neural architectures (e.g., Vaswani et al., 2017) and learning algorithms (e.g., Tan et al., 2018; Rennie et al., 2017), incorporating richer problem structures (e.g., structures between $\boldsymbol{x}$ and $\boldsymbol{x}'$) and linguistic knowledge through learning constraints (Hu et al., 2018; 2016) and structure bias (Strubell et al., 2018; Yang et al., 2016), as well as better leveraging the reference sentence as a template through mask-and-infilling (Zhu et al., 2019) (much like the rule-based approach). It is also interesting to extend from generating a single sentence to generating a whole passage (e.g., a game report or a news article) with any desired writing style (Wiseman et al., 2017).

It is difficult to mathematically define text style, content, and the boundary between them. In the task of text style transfer, a style (or attribute) has to be explicitly defined (usually as a categorical variable), which can be difficult when it comes to abstract properties (e.g., word choices). In comparison, our setting of defining writing style based on a reference sentence provides an alternative, arguably more natural way of specifying the style of interest. This resembles the style definition in image style transfer (Gatys et al., 2016; Johnson et al., 2016) where visual style is specified with a reference stylistic image.

Besides the direct practical use of the task itself, the objective of preserving writing characteristics of a given reference sentence provides a way of preventing a data-to-text model from repeatedly generating generic, low-diversity text (Guu et al., 2017; Li et al., 2016). Generation diversity is explicitly controlled by the richness of reference sentences.

References

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. Evaluating the state-of-the-art of end-to-end natural language generation: The E2E NLG Challenge. *arXiv preprint arXiv:1901.11528*, January 2019.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414–2423, 2016.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. Incorporating copying mechanism in sequence-to-sequence learning. *ACL*, 2016.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. Generating sentences by editing prototypes. *arXiv preprint arXiv:1709.08878*, 2017.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. Generating sentences by editing prototypes. *Transactions of the Association of Computational Linguistics*, 6:437–450, 2018.

Tatsunori B Hashimoto, Kelvin Guu, Yonatan Oren, and Percy S Liang. A retrieve-and-edit framework for predicting structured outputs. In *Advances in Neural Information Processing Systems*, pp. 10073–10083, 2018.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing deep neural networks with logic rules. In *ACL*, 2016.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *ICML*, 2017.

Zhiting Hu, Zichao Yang, Ruslan R Salakhutdinov, LIANHUI Qin, Xiaodan Liang, Haoye Dong, and Eric P Xing. Deep generative models with learnable knowledge constraints. In *Advances in Neural Information Processing Systems*, pp. 10522–10533, 2018.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pp. 694–711. Springer, 2016.

Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 329–339, 2016.

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

Rémi Lebret, David Grangier, and Michael Auli. Neural text generation from structured data with application to the biography domain. *arXiv preprint arXiv:1603.07771*, 2016.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *NAACL*, 2016.

Juncen Li, Robin Jia, He He, and Percy Liang. Delete, retrieve, generate: A simple approach to sentiment and style transfer. *arXiv preprint arXiv:1804.06437*, 2018.

Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. Content preserving text generation with attribute controls. In *Advances in Neural Information Processing Systems*, pp. 5108–5118, 2018.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

Feng Nie, Jinpeng Wang, Jin-Ge Yao, Rong Pan, and Chin-Yew Lin. Operations guided neural networks for high fidelity data-to-text generation. *EMNLP*, 2018.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. Style transfer through back-translation. *arXiv preprint arXiv:1804.09000*, 2018.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7008–7024, 2017.

Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. *ACL*, 2017.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *ACL*, 2015.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, pp. 6830–6841, 2017.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. Linguistically-informed self-attention for semantic role labeling. In *EMNLP*, 2018.

Sandeep Subramanian, Guillaume Lample, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. Multiple-attribute text rewriting. In *ICLR*, 2019.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NeurIPS*, pp. 3104–3112, 2014.

Bowen Tan, Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric Xing. Connecting the dots between MLE and RL for text generation. 2018.

Youzhi Tian, Zhiting Hu, and Zhou Yu. Structured content preservation for unsupervised text style transfer. *arXiv preprint arXiv:1810.06526*, 2018.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *ACL*, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, pp. 2692–2700, 2015.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*, 2015.

Jason Weston, Emily Dinan, and Alexander H Miller. Retrieve and refine: Improved sequence generation models for dialogue. *arXiv preprint arXiv:1808.04776*, 2018.

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. Challenges in data-to-document generation. In *EMNLP*, 2017.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Zichao Yang, Phil Blunsom, Chris Dyer, and Wang Ling. Reference-aware language models. *arXiv preprint arXiv:1611.01628*, 2016.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric Xing, and Taylor Berg-Kirkpatrick. Unsupervised text style transfer using language models as discriminators. In *NeurIPS*, 2018.

Wanrong Zhu, Zhiting Hu, and Eric Xing. Text infilling. *arXiv preprint arXiv:1901.00158*, 2019.

A Data Creation

Our dataset is made from the RotoWire dataset (Wiseman et al., 2017), which consists of paired tables and paragraphs, each describing a NBA game. Each table is consisted of data records about box- and line-scores of basketball games, while each corresponding paragraph written by specialists is summarizing the game. In order to obtain desired $(\boldsymbol{x}, \boldsymbol{y})$ pairs, we first separate the paragraphs into sentences. Each sentence will be our $\boldsymbol{y}$. However, it is not so easy to obtain the corresponding $\boldsymbol{x}$ (i.e. data records): we have to find out which data records in the table of data records are described in $\boldsymbol{y}$.

To achieve this, we proposed a not precise but accurate enough rule based method to get $\boldsymbol{x}$ from $\boldsymbol{y}$. Two steps are performed in our method: Step 1, find out all entities (player, team, and city) and numbers in $\boldsymbol{y}$; Step 2, pick out candidate (entity, number) pairs that will form our final $\boldsymbol{x}$.

In step 1, we first collect all team/player/city names as all entities in the training set. We locate these entities in $\boldsymbol{y}$ and replace every multi-word entity name in $\boldsymbol{y}$ by an underscore-connected single token in order to simplify the task. These tokens are then recognized as entities. To find out the numbers, we simply invokes a Python module called text2num (Copyed from exogen's text2num.py) which can recognize and convert every English numbers in $\boldsymbol{y}$ to digital numbers. Then all digital numbers are recognized.

In step 2, we enumerate every pair of (entity, number) found in step 1 and retrieve all candidate data records from the table. We simply iterate over the table and pick out those records whose entity name and score are exactly the same as the entity and number in our pair. However, there can be multiple records picked out which have different or even contradict labels due to the ambiguity. To reduce wrong records retrieved, we add more rule constraints to filter out obviously wrong records. For instance, if the succeeding token of the number is 'assist', then those records with labels related to 'rebound' or 'turnover' are obviously wrong. After this, there is still some redundancy in these records, though, we believe it is accurate enough to collect these records as $\boldsymbol{x}$. Finally, the entity names are also added to $\boldsymbol{x}$ so that the copy mechanism can directly copy the entity names from $\boldsymbol{x}$.

After obtained all $(\boldsymbol{x}, \boldsymbol{y})$ pairs of the training/validation/test sets, we now have to assign a $(\boldsymbol{x}', \boldsymbol{y}')$ to each of them. The $(\boldsymbol{x}', \boldsymbol{y}')$ is always from the training set, even for $(\boldsymbol{x}, \boldsymbol{y})$ in the validation/test sets.

The $(\boldsymbol{x}', \boldsymbol{y}')$ should meet our requirement that $\boldsymbol{x}$ and $\boldsymbol{x}'$ are quite similar but not completely match in their labels. We define a match function to calculate the similarity between $\boldsymbol{x}$ and all $\boldsymbol{x}'$ in the training set. The match function is: we first define a similarity score between every pair of records between $\boldsymbol{x}$ and $\boldsymbol{x}'$, then use Kuhn-Munkres algorithm to find out the best match of the bipartite graph with scores as weights. The sum of scores in the match is our similarity of $\boldsymbol{x}$ and $\boldsymbol{x}'$, which is the output of our function. The $\boldsymbol{x}$ - $\boldsymbol{x}'$ match also act as an crucial part in our rule based model described above.