Generating Accurate Caption Units for Figure Captioning

Xin Qian*
University of Maryland, College Park
xinq@umd.edu

Eunyee Koh
Adobe Research
eunyee@adobe.com

Fan Du
Adobe Research
fdu@adobe.com

Sungchul Kim
Adobe Research
sukim@adobe.com

Joel Chan
University of Maryland, College Park
joelchan@umd.edu

Ryan A. Rossi
Adobe Research
ryrossi@adobe.com

Sana Malik
Adobe Research
sana.malik@adobe.com

Tak Yeon Lee*
Dept. of Industrial Design, KAIST
reflect9@gmail.com

ABSTRACT

Scientific-style figures are commonly used on the web to present numerical information. Captions that tell accurate figure information and sound natural would significantly improve figure accessibility. In this paper, we present promising results on machine figure captioning. A recent corpus analysis of real-world captions reveals that machine figure captioning systems should start by generating accurate caption units. We formulate the caption unit generation problem as a controlled captioning problem. Given a caption unit type as a control signal, a model generates an accurate caption unit of that type. As a proof-of-concept on single bar charts, we propose a model, `FigJAM`, that achieves this goal through utilizing metadata information and a joint static and dynamic dictionary. Quantitative evaluations with two datasets from the figure question answering task show that our model can generate more accurate caption units than competitive baseline models. A user study with ten human experts confirms the value of machine-generated caption units in their standalone accuracy and naturalness. Finally, a post-editing simulation study demonstrates the potential for models to paraphrase and stitch together single-type caption units into multi-type captions by learning from data.

KEYWORDS

Data visualization, web accessibility, text generation, image captioning, figure question answering

ACM Reference Format:
Xin Qian, Eunyee Koh, Fan Du, Sungchul Kim, Joel Chan, Ryan A. Rossi, Sana Malik, and Tak Yeon Lee. 2021. Generating Accurate Caption Units for Figure Captioning. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3442381.3449923

*Work performed while at Adobe Research.

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449923

1 INTRODUCTION

Scientific-style figures are important media forms to present numerical information in a wide spectrum of documents (e.g., HTML, PDF). Captions are pieces of text accompanying figures that summarize their information. Accurate and natural-sounding captions can improve the accessibility and usefulness of figures and documents. For instance, captions can help readers quickly grasp a web page's main ideas during skim reading. Captions can scaffold as the alternative text of figures for users with visual impairments [52] and users with low network bandwidth for loading figures. A detailed caption can also increase the retrievability of a web page by search engine crawlers. Unfortunately, descriptions for figures in documents are often trivial, non-informative, or absent altogether [6]. The increased calls for web accessibility and new tools like automated accessibility checkers [9] lead us to ponder the possibility of automating caption generation. Such systems could provide much value when integrated into existing web publishing tools such as WordPress, Google Web Designer, and Adobe Acrobat.

To automatically generate captions for figures, the machine needs to parse figure elements, reason over the relationships between elements, then describe the relationships in natural language. Recent advances in more general vision–language problems such as visual question answering [1, 33] and image captioning [51, 54] show how well the machine can reason about and describe an image. Closer to our application domain, work on figure question answering [28, 29] and figure element extraction [44, 49] demonstrates machine capability on scientific-style figures.

However, it is unclear how the outputs of these approaches meet the goal of accessibility. For example, figure question answering [28, 29] assumes that users generate questions about a figure. However, visually impaired users may need some descriptions of the figure before asking questions. Figure element extraction [44, 49] also does not answer the residual question of how users should interpret parsed figure elements to understand the figure. Taking a leap from previous applications, we aim to generate captions useful to users but with minimum requirements on user interaction, making it a "last mile" problem with directly applicable value for end-users.

A recent corpus analysis on human-written figure captions from the IELTS English Language Test [45] finds that caption paragraphs are composed of separable caption units. The caption units are clauses of a finite set of types (e.g., number and labels of items,

pairwise comparisons) that describe specific types of information in figures. This discovery of caption units suggests that we can frame the overall problem of figure captioning as a multi-stage inquiry. The first stage can focus on the problem of generating accurate caption units of specific types. Meanwhile, subsequent stages can focus on stitching the caption units into overall caption paragraphs. In this paper, we follow by focusing on the first stage problem of automatically generating accurate caption units. We formulate the task as a controlled image captioning problem: given a control signal of the caption type, a model generates a caption unit of that type. While real-world observations bring forth the concept of caption units, there are no readily available, large-scale datasets for doing supervised machine learning on this task. We create datasets of figures and caption units on single bar charts from two existing figure question answering datasets: DVQA [28] and FigureQA [29].

We propose the FigJAM (**Fig**ure captioning with **J**oint **A**ttention to **M**ulti-modal information) model for single bar charts. It generates accurate caption units for a given control signal by jointly attending to multi-modal information. Attention is a neural network mechanism that re-weights relevant features from the encoding side to improve generation quality [54]. Multi-modal information of a figure includes both raw figure image and metadata. Image metadata has been shown effective in disentangling visual reasoning tasks from general image understanding [58]. Attending directly to metadata information helps FigJAM achieve better visual-semantic alignment [30] than baselines, [1] an ability to align a visual figure element with the element's underlying name. For text labels in the figure that are out-of-vocabulary (OOV) words in general English, FigJAM incorporates a dynamic dictionary, following the design of the DVQA figure question answering model [28].

To validate the effectiveness of the problem formulation and FigJAM, we conduct quantitative evaluations with the two datasets. Experiment results show that by incorporating metadata information and dynamic dictionary, FigJAM can generate more accurate caption units over competitive baselines. The baselines include a general CNN-LSTM-Attn model, an FCAP model with pixel-based relation network and partial utilization of metadata information, and a DVQA-adapted captioning model with the dynamic dictionary. A user study with ten human experts further confirms the value of machine-generated caption units in their standalone accuracy and naturalness. Finally, a post-editing simulation study on two suitable datasets converted from DVQA and LEAF-QA [10, 28] demonstrates that our proposed model can paraphrase and stitch single-type caption units into multi-type captions.

In summary, this paper makes the following contributions:

- **Problem Formulation (Sec. 3):** We formulate the problem of generating accurate caption units as a controlled image captioning problem.
- **Model (Sec. 4):** We introduce FigJAM, a model that utilizes metadata information and incorporates a figure-specific dynamic dictionary to tackle the problem on single bar charts.
- **Evaluation Results (Sec. 5 – 8):** We analyze the effectiveness of the problem formulation and the FigJAM model through quantitative evaluations in metrics, a user study, and a post-editing simulation study.

[1] In general, the concept of visual-semantic alignment [30] refers to connecting image objects with their valid semantic meanings outside of the images. Here, we operationalize the concept as the ability to connect figure elements (e.g., a bar) with their text labels. Captions mention text labels to refer to the elements.

2 RELATED WORK

Our work builds on ideas from three related problems.

2.1 Image Captioning

Most recent work on image captioning employs an end-to-end, neural encoder-decoder structure [30, 51] with the attention mechanism [54], which gives a top-down, general description of the image. A few approaches make general captions more controllable and specific, including having different styles (sentiments, attractiveness, etc.) [20, 37], or being grounded with bottom-up semantic concepts (attributes, entities, objects, etc.) [3, 7, 15, 57, 59]. Dense captioning [27, 32] and dense relational captioning [31] generate a plurality of captions for one single image (object existence, relationships, etc.). Figure captioning has exclusive challenges, including caption accuracy and multi-hop visual-semantic alignment. We formulate the task as a controlled image captioning problem. Generated caption units could further group into a dense caption.

2.2 Figure Question Answering

Visual question answering (VQA) is the task of answering questions about images from either real-world scenes [1, 33, 35] or synthetic scenes [1, 26]. Figure question answering (FQA) tackles question answering on scientific-style figures. DVQA [28] and FigureQA [29] are two public datasets on FQA. LEAF-QA [10] is another recently proposed dataset. FQA has some unique challenges over usual VQA problems [28], including proper handling of figure-specific vocabulary and visual-semantic alignment. Our problem of figure captioning differs from FQA since no question is asked from the user. This work aims to bridge the gap between FQA and spontaneous figure captioning.

2.3 Parsing Figures & Rule-based Captioning

A body of related work has studied the problem of parsing and reconstructing figures [13, 44, 49]. The work focuses on the task of extracting visual elements and mapping them into a predefined data structure. However, they do not address the problem of generating human-readable descriptions for the elements [23]. Several corpus studies [17, 18] examine the communicative goals behind infographic captions and provide ideas for automatically generating such captions. This thread of work creates a taxonomy of intended messages to classify each caption sentence into an intent category [38]. A recent corpus analysis study [45] examines captions with multiple sentences from a real-world corpus. The study presents design guidelines for generating captions with multiple sentences to communicate intended messages.

Rule-based techniques have been widely applied for generating descriptive text and figure captions. Mittal et al. [39] design a rule-based text planning system that describes graphics generated by SAGE-system [46]. PostGraphe [19] and SelTex [14] are two other rule-based text generators, focusing on text generation with access to underlying data in tabular form. Several systems also propose

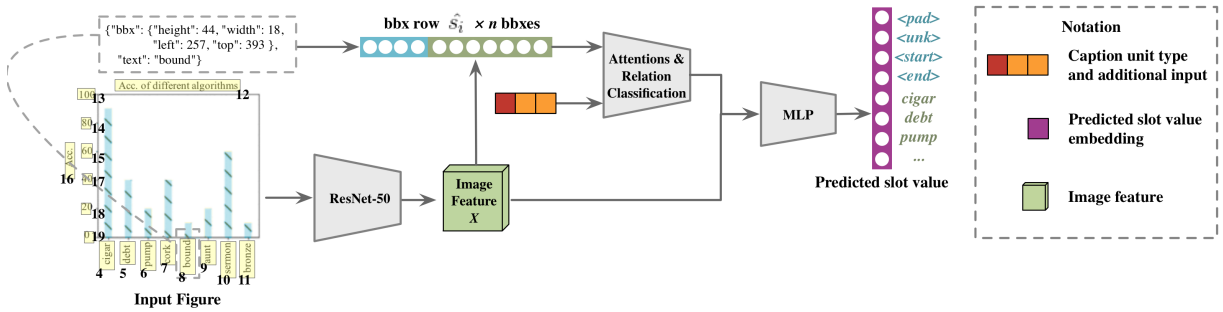


Figure 1: Model architecture of FigJAM (Sec. 4.1.1–4.1.5). The left-most are an input figure and metadata in yellow boxes. Numbers on yellow boxes are indexes in the dynamic dictionary. The right-most purple vector is the predicted slot value over a joint static and dynamic dictionary.

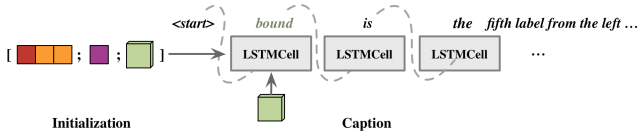


Figure 2: Model architecture of FigJAM (Sec. 4.1.6).

modularized pipelines for summarizing line graphs and bar charts with texts [16, 40]. These pipelines usually consist of an information extraction module, an intent recognition module, and a text planning module. Follow-up research proposes cluster-based [4] or semantic graph [2] approaches to enhance the pipelines. Compared to these rule-based generators, FigJAM is more flexible at reasoning over arbitrary pixel figures and modeling language.

3 PROBLEM FORMULATION

We formulate the problem of generating caption units as a controlled image captioning problem, where a model generates a caption unit specific to a caption type. Illustrated on the left of Figure 1, let an input pixel figure image X, having the metadata information, be denoted as $\mathcal{T}$. Here, we assume having the metadata information in the simplest level, which are OCR-extracted text labels and their bounding boxes. Therefore, $\mathcal{T}$ represents a set of tuples

$$\mathcal{T} = \{(s_i, c_i)\}_{i=1}^{n} = \{(s_1, c_1), (s_2, c_2), ..., (s_n, c_n)\} \quad (1)$$

Now, given a caption type t (e.g., *title*, *figure type*, *count*, *label name*), the goal for the model is to generate a typed caption sentence C, which we call the *caption unit*, for the input figure X representing a sequence of words $C = (w_1, w_2, \ldots, w_l)$, where $l = |C|$ is the length of the caption unit that varies. Each tuple $(s_i, c_i) \in \mathcal{T}$ contains an alphanumeric text label s_i and bounding box coordinates c_i of s_i. n is the number of text labels in the figure. A figure satisfies that $n \geq 1$, i.e., containing at least one alphanumeric text label. Optionally, caption units of specific types may require additional inputs $d = \{d_1, ..., d_m\}$ as guiding signals, where d is m pieces of information to nail down a unique ground truth. We list the required additional inputs in the third column of Table 1. Suppose the caption unit C is "*Cage is the label of the first bar from the bottom*"; then, the additional input d here will be the ordinal number of a bar, 1, to make sure a caption unit describes the name of a particular bar (the first bar). Having a unique ground truth ensures the correct measurement of caption accuracy. Putting everything together, given a corpus $\mathcal{D} = \{X, \mathcal{T}, t, d, C\}_{i=1}^{N}$, the goal is to learn a captioning model $\mathcal{M}$ that generates caption units. Once the captioning model $\mathcal{M}$ is learned, we can then use it to infer a caption unit for a new unseen figure. More formally, given $\mathcal{M}$ along with a new figure $(X, \mathcal{T})$, we generate caption units where each of them has one specific type t with corresponding additional inputs of d.

4 PROPOSED MODEL: FIGJAM

We propose the FigJAM (**Fig**ure captioning with **J**oint **A**ttention to **M**ulti-modal information) model. It is suitable for generating caption units on single bar charts.

4.1 Model Architecture

Figure 1 and 2 shows the model architecture of FigJAM.

4.1.1 Encoding input. On the left-most are four parts of input to the model: (1) a pixel figure image; (2) a set of OCR-extracted bounding box tuples as the metadata information (the gray dotted box); (3) the caption unit type (the red vector); (4) the additional input required for the type (the orange vector). As in Table 1, additional inputs can be ordinal numbers or elements. We specify them in the vector as the word indexes of ordinal numbers or elements from a joint dynamic and static dictionary. Following ideas from DVQA [28], the dynamic dictionary consists of 30 reserved word indexes to accommodate unique text labels from OCR-extracted bounding boxes in a figure. The same word index has different denotations in different figures. A CNN (e.g., ResNet-50) encodes the raw figure image to get its features as $X = \{x_{1,1}, ..., x_{H,W}\} \in \mathbb{R}^m$, where H and W are the height and width from CNN output, and m is the number of feature maps. The image feature is useful for calculating (along with the metadata information) the joint attention for slot value prediction and the adaptive soft-attention at each decoding time step.

We encode the caption type as a vector t of length 1. We represent additional input associated with a caption type as a vector $d = \{d_1, ..., d_{m'}\}$ where m' is the maximum number of additional inputs across all types. For caption unit types that do not require additional input, d is a zero-filled vector of the same length.[2] We concatenate

[2] Alternatively, the caption unit type t could be encoded as an embedding representation. We did not do this since the number of caption unit types is small—a maximum of seven types in the current setting.

Table 1: Summary of caption types, descriptions, additional inputs, and example outputs.

Caption type	Description	Additional inputs	Example output (with **slot value**)
Title	Mentioning the title, usually with paraphrasing.	None	This table shows **sales statistics for items in different stores.**
Figure type	High-level figure type (horizontal/vertical, bar/pie/line, etc.	None	This is a **horizontal** bar chart.
Count	Counting the number of elements in the figure.	None	There are **three** bars in the chart.
Label name	Describing the text label name for an ordered position.	Ordinal number Nth	**Cage** is the label of the first bar from the bottom.
Min/max	Describing the element that has the minimum/maximum value.	None	**Frame** has the highest accuracy.
Comparison	Describing the comparative advantage of one element over the other.	Element *X, Y*	The accuracy of the algorithm damage is **larger** than ward.
Value	Describing the value of an element.	Element X	**80** is the accuracy of the algorithm brace.

Table 2: Dataset statistics for each caption unit type. Note that FigureQA-cap data does not have *title* or *value* types.

Dataset	Split	# Images	Title	Figure type	Count	Label name	Min/max	Comparison	Value
FigureQA-cap [29]	train	37,000	N/A	37,000	37,000	214,327	33,499	148,000	N/A
	validation	3,000	N/A	2,370	2,504	2,323	2,299	2,678	N/A
	test_easy	3,000	N/A	2,438	2,583	3,063	2,692	2,458	N/A
	test_hard	3,000	N/A	2,315	1,989	2,731	2,689	2,558	N/A
DVQA-cap [28]	train	195,000	195,000	115,516	39,935	194,426	61,847	67,867	20,379
	validation	5,000	2,838	3,240	1,033	2,908	1,571	1,700	533
	test_easy	5,000	2,934	2,606	961	2,950	1,563	1,763	510
	test_hard	5,000	2,451	2,997	1,046	3,152	1,603	1,704	482

caption type vector and additional input vector as the query vector (red and orange color block in Figure 1) $\hat{d} = [t; d]$.

Given the set of metadata information, we construct a matrix $S \in \mathbb{R}^{n \times 6}$, where n is a figure-specific value denoting the number of alphanumeric text labels in the figure. 6 is the column dimension of S where each row s'_i consists of (1) the metadata coordinates c_i; (2) a dynamic word index associated with the text label s_i; (3) a binary bit indicating whether the text label contains only digits.

4.1.2 Attention to positional ordering (the blue vector at the top). The caption unit type *label name* describes the text label name for a given position. It is necessary to infer from ground-truth text label's coordinates as well as its relative value to the coordinates of other text labels.

Therefore, we calculate the first attention by using the query vector $\hat{d}$ (which contains the ordered position) to query the bounding box matrix S. Specifically, let s'_i be the i-th row in S. The attention weight to this row calculates as follows. First, the interaction between the row and the query vector is captured by an MLP.

$$e_i = Attn\,(s'_i, \hat{d}) = v^\top tanh\,(Ws'_i + U\hat{d}) \tag{2}$$

Outputs for each bounding box row aggregate into a matrix and then go through another fully-connected MLP layer to sort out a relative ordering of the bounding boxes ("sorting MLP"). A softmax function follows the MLP. Doing softmax gives the final attention weights to each bounding box row.

$$a = [a_0; a_1; ...; a_n] = softmax\,(MLP\,([e_0; e_1; ...; e_n])) \tag{3}$$

4.1.3 Attention to object-based value (the green vector at the top). The caption unit type *min/max* describes the element that has the min/max value. It is necessary to compare values (e.g., bar heights) associated with each text labels. For orthogonal figure types like bar charts (as opposed to, e.g., pie charts), the coordinates of text labels serve as an anchor to know the location of the elements (e.g., bars) in the figure.

Therefore, we augment the bounding box matrix S into $\hat{S}$. For each row s'_i in S that corresponds to one bounding box text label, we append it with local features of the same row and the same column as its coordinate from the image feature X. The augmented features work as local, hard attention to the figure, guided by object-specific information.

Let s'_i be the i-th metadata rows in S. Based on Sec. 4.1.1, it includes the metadata coordinates $c_i = (\mathring{l}, \mathring{t}, \mathring{w}, \mathring{h})$ where $\mathring{l}$ is the distance of the bounding box to the left margin, $\mathring{t}$ is its distance to the top, $\mathring{w}$ is its width, and $\mathring{h}$ is its height. The row position of the bounding box is $\mathring{r}_i = \mathring{t} + \frac{\mathring{h}}{2}$ while the column position is $\mathring{c}_i = \mathring{l} + \frac{\mathring{w}}{2}$. The augmented vector for the bounding box then becomes

$$\hat{s_i} = [s'_i; (x_{\mathring{r}_i,1}, ..., x_{\mathring{r}_i,W}); (x_{1,\mathring{c}_i}, ..., x_{H,\mathring{c}_i})] \tag{4}$$

Similar to the first attention, the attention weights for each bounding box information calculate as

$$\hat{e}_i = Attn\,(\hat{s_i}, \hat{d}) = v^\top tanh\,(W\hat{s_i} + U\hat{d}) \tag{5}$$

$$\hat{a} = [\hat{a}_0; \hat{a}_1; ...; \hat{a}_n] = softmax\,([\hat{e}_0; \hat{e}_1; ...; \hat{e}_n]) \tag{6}$$

4.1.4 Relation classification on object-based value pairs. The *comparison* type describes the comparative advantage of one element over the other. Inspired by the relation network [47] that models relations between object descriptor boxes on the CLEVR dataset [26], we introduce a classification component that evaluates the relations between figure elements through metadata information. For *comparison*, as the query vector $\hat{d}$ specifies the two objects to be compared, $\hat{d}$ is used to retrieve two bounding box rows from $\hat{S}$, namely $\hat{s_i}$ and $\hat{s_j}$. The subtraction between $\hat{s_i}$ and $\hat{s_j}$ goes through an MLP, which gives the result for relation classification.

$$\tilde{a} = MLP\,(\hat{s_i} - \hat{s_j}) \tag{7}$$

This relation modeling is different from previous work on figure question answering [29]. The latter models pixel-level pairwise

relation. In Sec. 6.2.2, we show that pixel-level is too fine as the granularity and inefficient for object-wise comparison.

4.1.5 Auxiliary slot value classification (the purple vector at the top right). We aggregate the image feature, attention weights, relation classification results and then pass them into an auxiliary classification module. The module predicts a dictionary word among the static and dynamic dictionary as the slot value word w (Sec. 5.1.3) before generating a caption unit. The classification module uses an MLP whose output is a vector of the same size as the joint dictionary. More formally,

$$w = MLP([a; \hat{a}; \tilde{a}; X]) \tag{8}$$

4.1.6 Initializing the decoder LSTM. As shown in Figure 2, After obtaining the slot value word w, we concatenate four sources of information as a joint vector: (1) the caption unit type vector t; (2) the embedding of additional inputs d; (3) the embedding of the slot value word w; (4) the image feature X. The joint vector initializes the decoder LSTM. At each step, the decoder LSTM predicts with attention weighting to the image feature and the word embedding of the last predicted word [54].

5 QUANTITATIVE EVALUATION SETUP

This section describes the setup for quantitative evaluations on FigJAM, where we create captioning datasets for single bar charts, and design metrics and baselines.

5.1 Data Preparation

Table 1 defines and exemplifies each caption type. There are four steps in data preparation: creating caption units (Sec. 5.1.1), incorporating metadata (Sec. 5.1.2), defining slot value (Sec. 5.1.3) and additional inputs (Sec. 5.1.4). Our dataset is available at this link.

5.1.1 Creating Caption Units. The IELTS corpus analysis [45] presents a set of dominant caption types. In this work, we aim to generate accurate caption units of a wide variety of caption types, including *title*, *figure type*, *count*, *label name*, *min/max*, *comparison*, and *value*. Unfortunately, there is no directly available supervised captioning dataset on those. Therefore, we create our datasets by making appropriate updates to figure question answering datasets. Both DVQA [28] and FigureQA [29] are large-scale question answering datasets on scientific-style figures. DVQA has over 3 million question–answer pairs for bar charts. FigureQA has more than 2 million question–answer pairs for five figure types. Both have two test splits, an easy one and a hard one, to test model generalizability over unseen semantics. From the corpus analysis, we find that these datasets contain a substantial amount of questions aligned with the set of caption unit types that we define.

Intuitively, combining and converting a question–answer pair of a question type into a statement sentence would yield a ground truth caption unit of that type. We nail down seven caption types to include in our datasets (as shown in Table 1). We use a SpaCy [24] POS tagger to convert the questions into statement sentences by following the wh-movement. Specifically, we replace each interrogative word with the actual answer and then re-order the sentence for grammatical soundness. Finally, two co-authors manually check the quality of converted captions by sampling 20 captions for each caption type in each of the two datasets. [3] All the sampled captions are accurate and grammatically correct except for one caption "*The chart shows Title.*" Table 2 lists the data statistics of our converted captions. We name our caption dataset converted from DVQA as DVQA-cap, and the one converted from FigureQA as FigureQA-cap.

There are several minor limitations in creating caption units of these types. First and foremost, we only created caption units for single bar charts. Although single bar charts are a subset of all figure types, we believe the problem formulation and methodology of this work would inspire approaches for stacked, group bar charts, and other figure types. Next, the *title* type is absent from FigureQA. For the *value* type, we only convert questions on figure elements whose value equals a quantized tick label. *Count* is not a dominant type in the corpus analysis.

5.1.2 Incorporating Metadata. Metadata extracted from the image has been shown effective in visual reasoning tasks [58]. Nowadays, extracting figure metadata has been a task with reasonable accuracy [25]. The status quo allows FigJAM to incorporate metadata information for attention, which offloads the burden to reason only from image feature. Consistent with the problem formulation (Sec. 3), for both datasets, we leverage the simple form of metadata in figures: OCR-extracted text labels and associated bounding box coordinates. Nevertheless, we believe that richer forms of metadata obtained from more complex metadata extraction techniques would further advance FigJAM and other caption unit generation model.

5.1.3 Slot Value. Inspired by the response generation task in task–oriented dialog systems [36], we define a slot value word for each caption type (see Table 1). The response generation task emphasizes the importance of correctly predicting a slot value within a response [34, 41] with the slot error rate metric [36, 53]. For caption unit generation, slot value correctness is also an important determinant [45]. The auxiliary classification module (Sec. 4.1.5) of FigJAM ensures correct prediction on the slot value and faster convergence rate.

Another benefit for defining slot values is using them as additional inputs to generate subsequent caption types, so as to form a caption paragraph. For example, *count* does not appear as a dominant caption type in the corpus analysis; we include it because its predicted slot value, the count of bars in the figure, can be the additional input to *label name*. After predicting 5 bars in the figure, we enumerate ordinal numbers from 1 to 5 and use each ordinal number as the additional input to generate a caption unit of the type *label name*. Similarly, the predicted slot values of *label name* pair with one another as the additional inputs to *comparison*. A planning algorithm could stitch together all caption units in Table 1 into a caption paragraph.

5.1.4 Additional Input. Previous work [12] trains and evaluates the captioning model on caption paragraphs with sequence-level metrics. This evaluation scheme obscures the level of caption accuracy. We advocate an accuracy-oriented evaluation on caption outputs, starting with a justification on why sequence-level metrics are not suitable. Consider an example ground truth caption paragraph "*This figure is a line plot. It contains three categories: yellow,*

[3] Quality check results are available at the same link.

magenta, sky blue ..." An inaccurate caption output of "*This figure is a dot plot. It contains three categories: teal, magenta, sky blue...*" would achieve a BLEU-4 [42] score of 0.356 and a METEOR [5] score of 0.372. Another accurate but paraphrased caption output "*There are three labels in this line plot. Their names are sky blue, magenta, yellow...*" would achieve a BLEU-4 score of 0 and a METEOR score of 0.313. The scores are lower than the inaccurate case. As an explanation, the family of sequence-level metrics (e.g., BLEU, METEOR) measures a general-sense similarity (instead of accuracy) between the ground truth and a generated caption output. It does not differentiate generated output errors between those in template language and truth slots. In the accurate but low-scored case, diversity in template language obscures an accurate caption.

Additional inputs guarantee a fair evaluation of the accuracy of each generated caption unit. Consider an alternative approach to evaluate is to include all accurate and possible caption units of the same caption type as ground truth. At evaluation time, a generated caption unit that matches any ground truth is considered accurate. This approach unavoidably introduces bias into the dataset, e.g., for *label name*, if caption units that discuss the first bar dominate, a trained model would adapt to such bias and be more prone to discuss the first bar. Instead, we define additional inputs associated with each ground truth caption unit to make the ground truth unique, an opposite characteristic to being diverse. A ground truth caption unit of type *label name* has the ordinal number Nth as its additional input. When $N = 3$, a generated caption that talks about "*Cage is the fourth bar from the bottom*" would be incorrect despite being truthful to the figure content. Therefore, introducing additional inputs aligns with the goal of accuracy-oriented evaluation.

5.2 Evaluation Metrics

We report accuracy (i.e., exact match rate, EM) and BLEU-4 scores.

$$Accuracy = \frac{\#\ of\ hypotheses\ matching\ references}{\#\ of\ references} \tag{9}$$

The fine-grained caption types and additional inputs guarantee that each example in the dataset has one unique ground truth caption. A predicted caption, when accurate, should exactly match with the ground truth.

We also include BLEU-4 [42] to cross-compare with numbers reported in FCAP [12]. However, as discussed in Sec. 5.1.4, sequence-level evaluation metrics are mainly suitable for general image captioning tasks in measuring meaningful variances between the hypotheses and the references. In our task, these differences undesirably obscure the level of accuracy.

5.3 Baselines

We evaluate three baselines in comparison with FigJAM. Table 4 lists the features of each model.

- **CNN-LSTM-Attn** [54]: a CNN-LSTM-based image captioning model with soft attention. The model uses a static dictionary of the model. For caption types with additional inputs, the additional inputs' embedding is appended to the image representation to initialize the LSTM and calculate the attention weights.
- **FCAP** [11, 12]: an extension to the CNN-LSTM-Attn with additional attention to the embeddings of text labels and pixel–based pairwise relations from the image features. It uses a static dictionary.
- **DVQA** [28]: a CNN-LSTM-based visual question answering model with stacked attention [56]. We adapt it to be a captioning model by replacing the output classifier with an LSTM decoder. We simplify the input question encoding into the input of a caption type and embeddings of additional inputs. It creates a dynamic dictionary.

5.4 Model Configuration and Implementation

For pre–processing, we resize figure images in all datasets and their corresponding metadata to be 448×448. To encode figure image, we fine-tune a pre-trained ResNet-50 up to the second last average pooling layer, the output image representation has a size of 14×14. We use the same approach as DVQA to construct the dynamic dictionary. Words that appear in the captions but are not text labels in the metadata form the static dictionary. The dynamic dictionary significantly reduces the dictionary size. On DVQA-cap, the dynamic and static dictionary gives a total size of 93. FigureQA-cap has 72. [4] Dictionary words are represented by 128-dimensional word embeddings, compatible with the reduced dictionary size. The auxiliary classification module is a two-layer MLP with an intermediate dimension of 512 and an output dimension of the dictionary size. Decoder LSTM has a hidden representation dimension of 256. We use teacher forcing during training and beam search with size one during testing. We use the Adam optimizer with a learning rate of $1e-3$ and cross-entropy loss. The loss applies to both the auxiliary classification module and the decoder. For training, we experienced longer convergence rates when jointly training for all caption types. Reported results are on separate training for each caption type. We trained the models on an NVIDIA Titan X graphics card for 50,000 batches with a batch size of 8.

6 QUANTITATIVE RESULTS AND ANALYSIS

Table 3 and Table 5 reports the accuracy of different models in generating caption units on the two datasets. Overall, our FigJAM model outperforms the competitive baselines, achieving high absolute scores (on the order of 90 and more) in terms of both accuracy and BLEU-4. This section compares FigJAM with the baselines and suggests where does FigJAM achieve its gains.

Despite the high values in the tables, our primary motivation is to present results of typing different caption units and demonstrate the technical bottleneck. For readers who might be interested in aggregated accuracy on the figure level, we include supplementary results at the same link. [5]

[4] With only the static dictionary, DVQA-cap has a dictionary size of 1,072; while FigureQA-cap has 175.

[5] We omit the results here for two concerns. First, the original DVQA and FigureQA models do not ask all possible (i.e., a complete set of) questions on each figure. Each figure in our DVQA-cap and FigureQA-cap datasets accordingly has an incomplete set of captions than the possible amount. As a remedy, we report overall accuracy (i.e., perfect accuracy) on 12 random figures through the user study (Sec. 7). Second,

Table 3: Accuracy and BLEU-4 scores on the FigureQA-cap dataset. Values are scaled by 100.

		Figure type		Count		Label name		Min/max		Comparison	
		Accuracy	BLEU-4	Accuracy	BLEU-4	Accuracy	BLEU-4	Accuracy	BLEU-4	Accuracy	BLEU-4
Test Familiar Color	CNN-LSTM-Attn	**100.00**	**100.00**	**100.00**	**100.00**	10.81	90.48	78.00	92.91	50.57	77.82
	FCAP	**100.00**	**100.00**	99.26	99.76	4.67	89.93	67.83	89.60	50.41	77.74
	DVQA	**100.00**	**100.00**	99.96	99.99	80.12	97.90	89.78	96.72	98.70	99.45
	FigJAM	**100.00**	**100.00**	99.96	99.99	**94.87**	**99.45**	**99.55**	**99.86**	**99.39**	**99.74**
Test Novel Color	CNN-LSTM-Attn	**100.00**	**100.00**	**100.00**	**100.00**	0.00	89.20	0.04	66.89	43.35	73.88
	FCAP	**100.00**	**100.00**	98.99	99.68	0.37	89.25	0.00	66.86	48.32	76.84
	DVQA	99.05	99.51	**100.00**	**100.00**	79.09	97.75	88.10	96.17	98.44	99.33
	FigJAM	**100.00**	**100.00**	**100.00**	**100.00**	**92.49**	**99.18**	**99.40**	**99.81**	**99.61**	**99.83**

Table 4: Features of baseline models and FigJAM.

	Dictionary	Directly using any metadata?	Relation network
CNN-LSTM-Attn	Static	No	No
FCAP		Text labels only	Pixel-based
DVQA	Dynamic	No	No
FigJAM		Text labels and coordinates	Object-based

6.1 Dynamic Dictionary vs. Static Dictionary

The group of models that use a static dictionary, CNN-LSTM-Attn and FCAP, performs consistently unwell for non-trivial caption types (*label name*, *min/max*, *comparison*, and *value*). The accuracy of these models is near the level of random guess: for *label name* and *min/max*, randomly recalling a correct word out of the static dictionary has a probability of smaller than 1%; for *comparison*, randomly guessing one of the *higher* or *lower* relations correct, without inferring from the two elements (specified by additional inputs), has a probability of near 50%; for *value*, the accuracy of 10–15% roughly means that the model tends to answer the most frequent value in the corpus. In contrast, models that use a dynamic dictionary, including DVQA and our FigJAM, are easily able to predict non-trivial caption types.

The comparative disadvantage of models with a static dictionary is that they cannot understand or re-describe a text label by its figure-specific context. They consistently associate a text label with its static meaning. The ability to associate an object in the figure with its semantic is called visual-semantic alignment [30]. Our problem operationalizes it as the ability to associate a figure element (e.g., a bar) with its name as in the text label. For instance, an element named *cigar* in figure *A* means its first bar while *cigar* in another figure B instead means its last bar. The only exception where models with a static dictionary (CNN-LSTM-Attn and FCAP) demonstrate this ability is when these models work on the test familiar color setting of the FigureQA-cap dataset. The accuracy for both models is relatively higher in Table 3 than in Table 5, approaching 10% for *label name* and over 50% for *min/max*). Both numbers are misleading due to potential feature leakage introduced from the original FigureQA dataset: the dataset uses figure element colors to name the elements. Correspondingly, models with a static dictionary learn to associate visual features extracted by CNN with color words in the dictionary. Unfortunately, this association does not generalize to the test novel color condition, nor does it generalize to the DVQA-cap dataset. At test time, we see examples where an FCAP model predicts a most similar in-vocabulary color name for the test novel color condition, such as "***midnight blue*** *is the label of ...*" instead of the correct caption of "***navy blue*** *is the label of ...*" Models with a static dictionary do not achieve real visual-semantic alignment since they cannot differentiate nuances in colors (*midnight blue* and *navy blue*).

Another motivation for having the dynamic dictionary is to handle text labels that are either multi-word units or out-of-vocabulary (OOV) words. The dynamic dictionary is similar to the copy mechanism in text generation tasks [22, 48]. At each decoder timestep, when a dynamic dictionary word has the highest probability, the model copies a multi-word unit into the output sequence. A caption unit of type *title* for the figure in Figure 1 can be "*The figure shows Acc. of different algorithms.*" Models with a static vocabulary would struggle to predict this caption unit, where "Acc." is likely an OOV word in the static dictionary. For corpora with a small vocabulary, where OOV is common, it is advantageous to have a dynamic dictionary.

6.2 Direct Utilization of Metadata Information

Here we analyze benefits for directly utilizing metadata information, which differentiates DVQA and FigJAM.

6.2.1 Handling positional ordering. FigJAM directly utilizes metadata information by encoding real-value coordinate information of text labels into the attention component that handles positional ordering (Sec. 4.1.2). The caption type of *label name* predicts the label name from a given ordered position (Nth). FigJAM uses the ordinal number Nth to query the fine-grained coordinates of all labels, followed by a "sorting MLP" module to get a relative ordering of these coordinates. The hypothesis is that the fully-connected "sorting MLP" module gives robust positional ordering by focusing on macro-level position differences while ignoring micro-level width or height differences between a column or a row of aligned text labels.

In contrast, DVQA does not directly model coordinate information. It uses the information indirectly to create the dynamic dictionary: DVQA applies an ordering heuristic on coordinate information to index text labels [28] as the dictionary. The heuristic is coarse-grained compared to coordinates' precision, making DVQA prone to predict the most frequent dynamic word index for a given position Nth instead. For example, when given an additional input of $N = 1$, the model tends to predict the dynamic word of index 4 since 4 is the

prior work shows limitations in evaluating aggregated captions using sequence-level metrics instead of accuracy. See the justification example in Sec. 5.1.4.

Table 5: Accuracy and BLEU-4 scores on the DVQA-cap dataset. Values are scaled by 100.

		Figure type		Count		Label name		Min/max		Comparison		Value	
		Accuracy	BLEU-4	Accuracy	BLEU-4	Accuracy	BLEU-4	Accuracy	BLEU-4	Accuracy	BLEU-4	Accuracy	BLEU-4
Test Familiar Vocabulary	CNN-LSTM-Attn	99.88	99.93	**98.02**	**99.45**	0.14	89.46	0.13	80.94	49.91	83.51	14.90	86.29
	FCAP	99.39	99.67	83.66	94.37	0.03	90.03	0.13	80.27	50.09	83.62	12.75	82.66
	DVQA	**99.96**	**99.98**	97.29	99.09	67.32	96.60	77.61	95.10	50.20	83.69	59.61	93.36
	FigJAM	99.81	99.87	96.98	98.84	**85.25**	**98.25**	**94.31**	**98.86**	**99.09**	**99.72**	**91.37**	**98.61**
Test Novel Vocabulary	CNN-LSTM-Attn	**99.93**	**99.96**	**97.71**	**99.39**	0.00	89.40	0.00	80.91	0.00	32.43	0.00	54.33
	FCAP	99.50	99.69	80.59	93.53	0.00	89.80	0.00	79.67	0.00	32.43	0.00	54.04
	DVQA	99.80	99.87	**97.71**	99.29	66.88	96.53	76.92	95.07	48.83	83.07	60.37	93.09
	FigJAM	99.63	99.75	95.79	98.41	**86.29**	**98.56**	**95.07**	**99.02**	**99.59**	**99.87**	**87.97**	**97.96**

most common index for words in the bottom left. Table 3 shows the results. Our approach significantly outperforms all other methods for non-trivial caption types (*label name*, *min/max*, and *comparison*) on both tasks (testing of familiar colors and novel colors) in Table 3.[6] CNN features in DVQA may help correct some bias between ordered position and dynamic word index, based on whether a figure uses a horizontal or vertical layout. Moreover, for label names in yellow bounding boxes as shown in Figure 1, small variances in figure's layout would affect the order of a dynamic dictionary. Variances include how tightly or loosely a figure positions label names along one dimension or even how it aligns those label names.

6.2.2 Object-based Attention. FigJAM has two object-based attention modules: one attends to object-based values (Sec. 4.1.3) and another to object-based value pairs for relation prediction (Sec. 4.1.4). Object-based attention gives FigJAM higher accuracy than DVQA on *min/max*, *comparison*, and *value*, by a large margin (+10–40%). The intuition behind utilizing metadata information (coordinates) for object-based attention is that coordinates of text labels directly point the model to know "where it should look at for an object" as well as "which two areas it should look at to compare two objects." We could see it as another instance of visual-semantic alignment. DVQA, without object-based guidance, relies solely on the image features and positional information from the dynamic dictionary index to navigate the right attention pattern over the entire figure image. The comparative advantage is similar to that of the relation network on state descriptions over CNN-LSTM with stacked attention on the CLEVR dataset [47]. One exception noted earlier is the high accuracy of DVQA (over 98%) for *comparison* on the FigureQA-cap dataset due to a systematic bias.

Comparing FigJAM with FCAP, the latter employs pixel-based relational attention for *comparison*. FCAP gives random relational prediction with near 50% accuracy while FigJAM performs surprisingly well. We argue that object-based attention for relation prediction is more effective than pixel-based one. First, it achieves visual-semantic alignment. When all figure elements (bars) have the same color and texture where visual features have no differences, pixel-based relational attention cannot differentiate the subject and object being compared. On the contrary, recognizing objects through coordinate anchors could separate the two figure elements being compared and capture the relative block size of these elements to achieve relational reasoning. Another drawback in pixel-based relational attention is a square increase in computation time and corresponding slow convergence: the model needs to compute the square of the number of pixels from the image feature. One explanation for the lower accuracy of FCAP for type *count* on DVQA-cap is that FCAP is slow at converging on this trivial type.

[6]One exception is the *comparison* caption type in Table 3, where the DVQA model performs competitively well (over 98%). One explanation is that the original FigureQA dataset is biased to only ask about the first two bars in view, according to its question generation script.

6.3 Generating Caption Units vs. Figure Question Answering

Since DVQA-cap and FigureQA-cap are both converted datasets from Figure Question Answering (FQA) [28, 29], one might expect to compare the accuracy between our caption generation task and FQA tasks. However, the accuracy of FigJAM and baseline models is not directly comparable to the classification accuracy in the FQA task. First, given a caption unit and the question–answer pair of the same type, spontaneously generating a caption unit is more challenging than understanding and answering a *yes/no* question. To answer the question "*Is aqua the maximum,*" a FQA model needs to recognize 5–10 contrasting colors that exist in figures and tell whether *aqua* is the highest among the few colors. To come up with a caption unit "*Aqua is the maximum,*" a captioning model needs to precisely recall, from all dictionary entries, the name of a color that is "halfway between blue and green" and not to mislead it with another similar color *cyan*. This comparison in task difficulty is analogous to Bloom's Taxonomy [8] of human cognitive level, where creating is at a higher level and more challenging than analyzing. Second, question–answer pairs included in the DVQA dataset are not the same as those included in the caption dataset. There is a question asking, "*Are figure values in a logarithmic scale?*" Moreover, the original DVQA work groups questions into three coarse-grained categories (*structure, data, reasoning*) and reports accuracy for each. Among the three categories, there are compound questions that combine multiple caption types in our definition. Lastly, the captioning task has a lower probability for a random guess to be correct. The DVQA dataset has questions such as "*Is the accuracy of **bound** lower than **cigar**.*" A random guess on these questions also gives a near 50% accuracy even when *bound* is an OOV word. In our problem, the two baselines that use a static dictionary have a near 0 accuracy in generating non-trivial caption units (see Table 5, *comparison* under "test novel vocabulary") such as "***bound** is lower than **cigar**.*" The models cannot correctly predict an OOV word *bound* as the subject.

6.4 Comparing Different Caption Types

Caption types vary substantially in their level of difficulty to generate. All models perform well on trivial caption types (*figure type* and *count*), since CNN alone can understand the overall figure structure. Non-trivial caption types (*label name*, *min/max*, *comparison*, and *value*) are more challenging. Caption units of the *value* type are currently limited to describing figure elements whose value corresponds to a quantized tick label on the axis. We wish to highlight that generating caption units of the *value* type resembles that of the *label name* type from a modeling perspective. The *label name* type requires the model to describe a label name on the axis based on an ordinal number. Similarly, the *value* type describes a value on the axis based on the named figure element's relative height.

7 USER STUDY

The previous section provides evidence of the efficacy of the controlled captioning problem formulation and discusses how key components of the FigJAM model (e.g., direct metadata utilization, dynamic dictionary) provide performance gains over baselines. This section presents an additional evaluation of the overall proposal, encompassing both the problem formulation and FigJAM outputs through a user study. While the fundamental importance is to establish the task of generating accurate caption units, the user study complements quantitative evaluations in Sec. 5 and 6 as additional validation. Presenting machine learning modeling outputs informs our readers, especially UX researchers, about the current technical landscape [55]. The goal is different from prior research focusing on either specific techniques [4, 11, 12, 14] or user scenarios in anticipation [6, 9, 43].

Despite that accessibility to the visually impaired population being a primary motivation for our work, this user study leaves openness to the actual use cases and definition of target end-users. [7] One future work is to depict more precisely the target end-users. Here we target the population as sighted participants, as they are a less vulnerable population at commenting on the correctness of the current emblematic system output. In the long run, we view sighted figure authors are a majority population that can actively contribute figure captions, e.g., editing from automatically generated captions. It is valuable to probe their perception of this task.

We recruit ten participants to write captions for and rate FigJAM-generated captions for 12 figures sampled from the DVQA dataset. All participants regularly write captions more than once a quarter (N = 5 write captions more than once per month; N = 2 more than once per week). The majority of our participants typically write captions for scientific papers and presentations, with a minority (N=2) having experience writing captions for accessibility reasons (e.g., alternative text) and general audiences.

Each participant reviews six figures sampled from the set of 12 figures; each set of six figures is drawn to cover the range of figure complexity in the DVQA dataset, as indicated by the number of bars in the chart (1 to 3, 4 to 7, or 8 to 10). In response to the call to evaluate aggregated accuracy (Sec. 6), one author manually creates missing ground truth caption units for the 12 figures, then re-runs the FigJAM model for machine output.[8] In total, each of the 12 figures is processed by an average of 5 participants (range from 2 to 8). For each figure, participants provide 7-point Likert-style ratings for three dimensions (quality, accuracy, or naturalness) of the overall caption (1 = very bad, inaccurate, or mechanical; 7 = very good, accurate, or natural), as well as naturalness ratings for caption units. To aid qualitative comparisons of caption quality and naturalness, we also ask participants to write captions of their own for the figures. Participants are free to use contents from the machine-generated captions for their captions. We also collect free-form comments for each caption.

Overall, participants rate the quality, accuracy, and naturalness of the machine-generated captions slightly above the midpoint of the 7-point scales. We interpret that participants judge the captions to be of reasonable quality, accuracy, and naturalness. Overall mean quality is 4.33 (SD = 1.31), while overall mean accuracy is 4.98 (SD = 1.61). For naturalness, overall mean naturalness is 4.19 (SD = 1.53), with slightly higher ratings for the naturalness of caption units, at a mean of 4.63 (SD = 1.89).

Qualitatively, participants make positive comments about caption accuracy. For example, P6 notes, *"I am surprised to see that your AI algorithm generates very accurate descriptions of the chart."* However, participants do detect errors in the captions for three of the 12 figures. This result gives an initial estimate for aggregated accuracy (Sec. 6). In one figure, the FigJAM-generated caption inaccurately notes that one of the bars had the highest value; in reality, the bar has the highest absolute value but is negative in amplitude compared to the other bars. In two other figures, machine-generated captions provide an inaccurate value for one bar.

Considering naturalness from a qualitative standpoint, a common theme in participants' comments is that the machine-generated captions as a whole tended to be more verbose and low-level than human-written captions. For example, P2 notes, *"machine generation captions is good, but in verbose style...if I am authoring my own captions, I would write more carefully, to make the captions sound succinct."* Similarly, P9 notes, *"The machine-generated captions are useful but a little bit too detailed."*

Note that participants are not instructed to write captions for a particular use case, and the majority of participants typically write captions for scientific publications, rather than the general audience or accessibility use cases. What seems to be missing in machine-generated captions for their typical use case is higher-level patterns, such as trends. For example, P4 notes that the machine-generated caption comprises "*trivial observations rather than synthesizing the pattern or interesting aspects of the figure.*" This issue is more salient for more complex figures with more bars, as P4 again notes, for a figure with nine bars, "*because there was a bigger number of bars than the previous figures, a large portion of the auto-generated caption was dedicated to trivial stuff.*" P1 makes a similar comment, noting that "*MG captions contain too many repetitions without creating higher-level insights.*" Figure 3 illustrates this point with two representative example figures from this study, comparing FigJAM-generated captions with human-written captions for the same figure. The examples show the overall accuracy and relative

[7] In Sec. 1, the problem of improving the accessibility and usefulness of figures has a broad context: busy skim readers, users with visual impairments or low network bandwidth for loading figures, and back-end search engines for figure retrievals.

[8] As this manual effort does not scale to whole datasets, we encourage the community to propose new datasets.

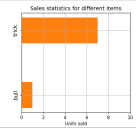
Figure	FigJAM-generated caption	Human-written caption
	The chart compares sales stats among different products. Two bars are there. The bars are horizontal. Bull is the label of the first bar from the bottom. Trick is the label of the second bar from the bottom. Trick, sold the most units. The item bull sold less units than trick.	1. The chart compares trick and bull unit in terms of the number of units sold. Trick, the first bar from top, was sold seven times. Bull, the second bar, was sold only one time. 2. The chart compares sales stats among different products. Trick sold more units than bull (7 vs. 1).
	The chart reports accuracy of different algorithms. Nine bars are there. The bars are horizontal. Wart is the label of the second bar from the bottom. Works is the label of the first bar from the bottom. Clue is the label of the fourth bar from the bottom. Soon is the label of the sixth bar from the bottom. Debt is the label of the eighth bar from the bottom. Map is the label of the seventh bar from the bottom. Bat is the label of the fifth bar from the bottom. Bid is the label of the third bar from the bottom. Bid, has the highest accuracy. The accuracy of the algorithm cult is smaller than bid. The accuracy of the algorithm map is smaller than cult. The accuracy of the algorithm cult is larger than bat. The accuracy of the algorithm wart is smaller than cult. The accuracy of the algorithm soon is smaller than cult. 2 is the accuracy of the algorithm works. 8 is the accuracy of the algorithm clue. 8 is the accuracy of the algorithm cult. 4 is the accuracy of the algorithm wart. 8 is the accuracy of the algorithm debt.	1. The chart reports accuracy of nine different algorithms. The Bid and bat algorithms have the highest accuracy at 9, followed by Cult, Debt and Clue at 8. The Map, Soon, Wart, Works algorithms' accuracies significantly lower than the rest. 2. The chart reports accuracy of nine different algorithms. There is a clear separation between high accuracy algorithms (cult, debt, bat, clue, and bid, each around 8 accuracy), and lower accuracy algorithms (map, soon, wart, and works, each around 1-2 accuracy).

Figure 3: Illustrative comparison between FigJAM-generated and human-generated captions for one lower complexity (top) and one higher complexity (bottom) figure from the user study. Note the overall accuracy and relative naturalness of FigJAM's caption units, but a verbose and slightly less natural overall caption paragraph than humans' condensed captions with higher-level patterns emphasized.

Table 6: Four illustrative post-editing rules. To prepare for the simulation experiment in Figure 4, we create the simulated human-written ground truth captions by rewriting original captions from each source pattern into one to two targets patterns using regular expressions.

DVQA-cap	Src	sales statistics for items in different stores
	Tgt 1	sales statistics for items **in each retail location**
	Tgt 2	sales statistics for **cokes in each CVS Pharmacy**
	Src	most preferred objects
	Tgt 1	**the most favorable** objects
	Tgt 2	**the highest ranked video games**
LEAF-QA-cap	Src	The chart shows $(.*)([0-9]\{4\})$ import
	Tgt	The chart **reports statistics for \\1 import in \\2**
	Src	The chart shows (.*) over Product_Code
	Tgt 1	The chart shows **product wise \\1**
	Tgt 2	The chart shows **\\1 over chocolate**

naturalness of FigJAM's caption units, but a verbose and slightly less natural overall caption paragraph compared to human-written condensed captions with higher-level patterns emphasized.

Notwithstanding, for some captioning use cases like accessibility, the balance of low-level to high-level information and the level of verbosity might be appropriately natural. For example, P6 comments that the more detailed, lower-level aspects of the captions "*would help increase not only readability but also accessibility of a figure..., such a captioning algorithm can help researchers create alternative texts for their charts if a figure-generating library employs it. Thus, such a library can create both a figure and its description, which can be used as an alternative text.*" Hence, verbosity may have some advantages when it comes to the scenario for visually-impaired readers because they may want to have as much information as possible, rather than having just condensed captions. We expect scientific use cases to drive modifications to the caption types in the problem formulation (e.g., including more higher-level caption units, removing lower-level caption units), rather than fundamental changes to the overall architecture or the modeling approach.

8 POST-EDITING SIMULATION

The quantitative experiments and the user study on FigJAM indicate that machine-generated captions have high accuracy but are not natural to humans. Prior IELTS corpus analysis highlights the same importance to generate natural captions [45]. One solution is to post-edit to improve caption naturalness. The term post-editing originates from machine translation, where humans modify machine-generated translation to improve translation quality [50]. To the accessibility use case for the visually-impaired population, post-editing a machine-generated caption would be valuable. While figures may have abbreviated text labels to avoid verbosity and save space, figure captions should avoid abbreviations and sound as detailed as possible. A screen reader that speaks an abbreviated word (e.g., "DLA") in captions like a regular English word ("dlah") would confuse the user. Finally, without post-editing, one may even argue whether FigJAM is more advantageous than a rule-based, pipelined figure captioning system [2, 14, 19, 40]. Given an accurate caption generated by FigJAM "*cage is the label of the first bar from the left,*" an alternative pipelined system could extract figure metadata and write decision rules to generate the same. Here we investigate the data-driven model FigJAM for handling two post-editing cases, based on realistic observations [45]: paraphrasing and stitching.

To achieve post-editing, let us revisit the FigJAM model from Sec. 4. FigJAM's dynamic dictionary handles cases when a figure text label is a multi-word unit. Specifically, when a multi-word unit is either an additional input or a predicted slot value word, its embedding from the dynamic dictionary is used to initialize the decoder. For the post-editing task, we also maintain a parallel static dictionary, in addition to the dynamic dictionary. Except that each multi-word unit appears as both an integral word in the dynamic dictionary and multiple tokenized words in the static dictionary. Whenever a multi-word unit appears as either the additional input (the orange block in Figure 1) or the predicted slot value word (the purple block), an additional LSTM encoder is used to encode its sequence, whose hidden state of the final timestamp is used to represent the multi-word unit. At decoding time, the encoded sequence's hidden state is used to initialize the decoder, instead of the dynamic dictionary word embedding of the multi-word unit. For clarity, we denote this slightly modified variant of FigJAM as FigJAM++. The effect of FigJAM++ looks like this: given a figure whose title is "*Import 2015 FYR Macedonia*," it generates a caption unit with abbreviation expanded such as

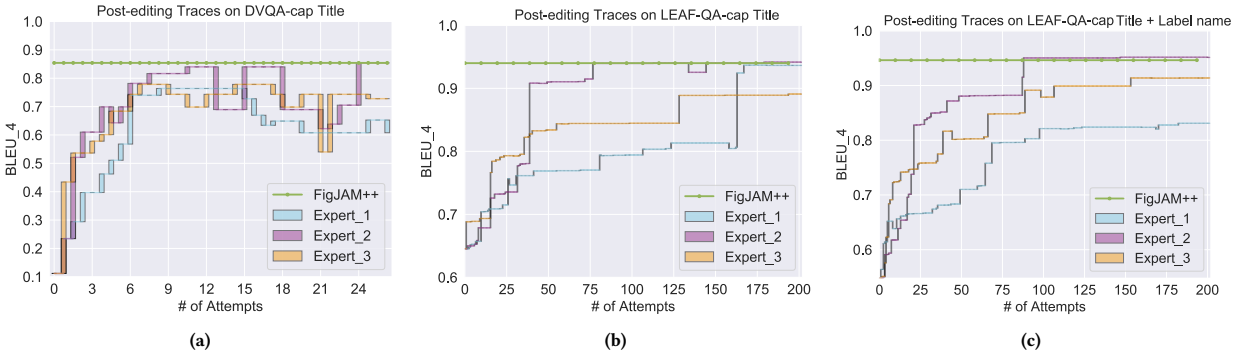


Figure 4: Simulated post-editing traces for rule-based, pipelined systems vs. FigJAM++ on DVQA-cap *title* (a); LEAF-QA-cap *title* (b); LEAF-QA-cap *title* + *label name* (c).

"*The figure shows imported value of the former Yugoslav Republic of Macedonia in 2015.*"

Inspired by simulation experiments in machine translation [21], the experiment in Figure 4 compares the naturalness of captions from a fully-converged FigJAM++ model with three simulated pipelined systems. For the latter, we hypothesize and simulate the scenario where domain experts manually inspect human-written caption corpora and write post-editing rules to tweak captions generated by pipelined systems. The research question is "*How many attempts would a domain expert need to make to improve a pipelined figure captioning system, such that its generated caption could sound as natural as a data-driven* FigJAM++ *model?*"

For the experiment, we identify two suitable datasets, DVQA [28] and LEAF-QA [10]. DVQA has seven unique titles that are multi-word units. LEAF-QA [10] is a more recent figure question answering dataset, whose major advantage is figures that visualize real-world data from sources including the U.S. Census and stock prices. The figures have meaningful, realistic text labels that are multi-word units. There are 2,624 unique titles, encompassing more diverse variations than DVQA. Similar to the process in Sec. 5.1.1, we first write rules based on figure metadata to construct a set of non-post-edited caption units for LEAF-QA, as the source for post-editing. We name the converted caption dataset from LEAF-QA as LEAF-QA-cap. FigJAM++ populates a joint dictionary of size 11,092 on LEAF-QA-cap, approaching the dictionary size of a realistic text corpus. We then select three caption types: *title* type on DVQA-cap, *title* type on LEAF-QA-cap, and *label name* on LEAF-QA-cap to simulate post-editing. For the last one, we revise the definition and ground truth of *label name* on LEAF-QA-cap to make the caption units into multi-type captions. We name it *title* + *label name*. To operationalize, the additional inputs of this type include the figure title. This revision tests the basic stitching capability for models to make single-type caption units into multi-type captions.

The simulation experiment starts with the three selected caption types and the non-post-edited ground truth captions of each type: DVQA-cap *title*, LEAF-QA-cap *title*, and LEAF-QA-cap *title* + *label name*. Two are single-type caption units and one is multi-type captions. We design seven rules for DVQA-cap and 50 rules for LEAF-QA-cap. The rules augment the datasets into parallel corpora with both the source, non-post-edited and the target, post-edited captions. Table 6 lists example post-editing rules. Each rule is a pair between one source pattern and at least one target pattern. Some rules have two target patterns to simulate two options for a domain expert to post-edit. One option is a primary post-editing pattern that we assume will be present in a corpus with a probability of 0.8 whenever the source pattern appears. Another is a secondary pattern with a probability of 0.2. The two options can be interpreted as two writing styles that co-exist in a corpus, e.g., general vs. context-specific, formal vs. informal, etc.

FigJAM++ is trained by taking figures and metadata as input, post-edited captions in mixed target patterns as ground truth output for each caption type until 50,000 batches. For pipelined systems, we first randomize (with replacement) the occurrences of post-edited captions by each target pattern. Then, we emulate three domain experts on their traces of iteratively recovering the rules from post-edited captions as they encounter, and tweaking a portion of outputs from FigJAM to improve caption quality using a new rule.

Due to the presence of primary and secondary post-editing patterns to the same source pattern with a probability distribution of [0.8, 0.2], FigJAM++ achieves accuracy scores of 0.792, 0.813, and 0.767, and BLEU_4 scores (flat green lines in Figure 4) of 0.854, 0.938, 0.932, respectively. These results suggest that FigJAM++ can learn from diverging patterns in a corpus and generalize to achieve reasonable accuracy. Among which, the caption type *title* + *label name* is multi-type. A reasonable accuracy in this type means the model is capable of learning basic stitching from data. While a converged FigJAM++ is having a stable BLEU-4 score, the BLEU-4 score of a simulated pipelined system through domain expert post-editing goes up when more rules are being discovered. In general, it takes at least two times the number of rules for the simulated expert to reach a similar caption quality to that of FigJAM++, e.g., only after approximately 14 attempts the expert boosts the caption quality for DVQA-cap *title*, as in Figure 4 (a). There are timestamps when a single rule in LEAF-QA-cap significantly boosts caption quality, which may be less likely in realistic scenarios. After reaching the highest quality, a domain expert may unconsciously revert a generalized primary rule to a rare secondary rule, as seen in the small drops of the traces below the green line. For a pipelined system with a domain expert that manually post-edits, reaching or maintaining the same level of caption quality as a data-driven model like FigJAM++ is challenging.

9 CONCLUSION AND FUTURE WORK

We formulate the caption generation problem as a controlled captioning problem: given a caption unit type as a control signal, a model generates an accurate caption unit of that type. As a proof-of-concept, we propose a new deep learning model, FigJAM, that utilizes metadata information and a joint static and dynamic dictionary. We conduct quantitative evaluations with two datasets from a related task of figure question answering. Results in accuracy and BLEU-4 show that FigJAM could generate more accurate caption units than competitive baseline models. A user study with ten figure-authoring participants confirms the value of machine-generated caption units, in their standalone accuracy and naturalness. Finally, a post-editing simulation study demonstrates our FigJAM's potential to paraphrase and stitch in improving caption naturalness.

Our work contributes towards generating accurate and natural figure captions, specially for scientific-style figures. Future work along the line includes stitching caption units together as a paragraph, extending the *value* type to describe arbitrary figure element values, and generalizing FigJAM beyond single bar charts to stacked and group bar charts, as well as other figure types. Like other deep learning research, our future work depends on appropriate dataset availability. Deployment is our long-term goal. A production system may also train from real users' captioning datasets. We hope our work can open up collaborations among the research community.

ACKNOWLEDGMENTS

The authors appreciate participants with the user study, as well as colleagues and anonymous reviewers for constructive feedback.

REFERENCES

[1] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C Lawrence Zitnick, Devi Parikh, and Dhruv Batra. 2016. VQA: Visual Question Answering. *International Journal of Computer Vision* 1, 123 (2016), 4–31.

[2] Rabah A Al-Zaidy, Sagnik Ray Choudhury, and C Lee Giles. 2016. Automatic summary generation for scientific data charts. In *Workshops at AAAI*.

[3] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*.

[4] Kasumi Aoki and Ichiro Kobayashi. 2016. Linguistic summarization using a weighted n-gram language model based on the similarity of time-series data. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 595–601.

[5] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

[6] Jeffrey P Bigham, Erin L Brady, Cole Gleason, Anhong Guo, and David A Shamma. 2016. An Uninteresting Tour Through Why Our Research Papers Aren't Accessible. In *CHI-EA*.

[7] Ali Furkan Biten, Lluis Gomez, Marçal Rusinol, and Dimosthenis Karatzas. 2019. Good News, Everyone! Context driven entity-aware captioning for news images. In *CVPR*.

[8] Benjamin S Bloom et al. 1956. Taxonomy of educational objectives. Vol. 1: Cognitive domain. *New York: McKay* (1956), 20–24.

[9] Erin Brady, Yu Zhong, and Jeffrey P. Bigham. 2015. Creating Accessible PDFs for Conference Proceedings. In *Proceedings of the 12th Web for All Conference*.

[10] Ritwick Chaudhry, Sumit Shekhar, Utkarsh Gupta, Pranav Maneriker, Prann Bansal, and Ajay Joshi. 2019. LEAF-QA: Locate, Encode & Attend for Figure Question Answering. *arXiv preprint arXiv:1907.12861* (2019).

[11] C. Chen, R. Zhang, E. Koh, S. Kim, S. Cohen, and R. Rossi. 2020. Figure Captioning with Relation Maps for Reasoning. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*.

[12] Charles Chen, Ruiyi Zhang, Eunyee Koh, Sungchul Kim, Scott Cohen, Tong Yu, Ryan A. Rossi, and Razvan C. Bunescu. 2019. Figure Captioning with Reasoning and Sequence-Level Training. *arXiv preprint arXiv:1906.02850* (2019).

[13] Daniel Chester and Stephanie Elzer. 2005. Getting computers to see information graphics so users do not have to. In *International Symposium on Methodologies for Intelligent Systems*. Springer, 660–668.

[14] Marc Corio and Guy Lapalme. 1999. Generation of texts for information graphics. In *Proceedings of the 7th European Workshop on Natural Language Generation*.

[15] Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. 2019. Show, Control and Tell: A Framework for Generating Controllable and Grounded Captions. In *CVPR*.

[16] Seniz Demir, Sandra Carberry, and Kathleen F. McCoy. 2008. Generating Textual Summaries of Bar Charts. In *INLG*.

[17] Stephanie Elzer, Sandra Carberry, Daniel Chester, Seniz Demir, Nancy Green, Ingrid Zukerman, and Keith Trnka. 2005. Exploring and exploiting the limited utility of captions in recognizing intention in information graphics. In *ACL*.

[18] Stephanie Elzer, Sandra Carberry, and Ingrid Zukerman. 2011. The Automated Understanding of Simple Bar Charts. *Artif. Intell.* 175, 2 (Feb. 2011), 526–555.

[19] Massimo Fasciano and Guy Lapalme. 1996. Postgraphe: a system for the generation of statistical graphics and text. In *INLG*.

[20] Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. 2017. Stylenet: Generating attractive visual captions with styles. In *CVPR*.

[21] David Grangier and Michael Auli. 2018. QuickEdit: Editing Text & Translations by Crossing Words Out. In *NAACL HLT*.

[22] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *ACL*.

[23] Braden Hancock, Hongrae Lee, and Cong Yu. 2019. Generating Titles for Web Tables. In *WWW*.

[24] Matthew Honnibal and Mark Johnson. 2015. An Improved Non-monotonic Transition System for Dependency Parsing. In *EMNLP*.

[25] Morten Jessen, Falk Böschen, and Ansgar Scherp. 2019. Text Localization in Scientific Figures using Fully Convolutional Neural Networks on Limited Training Data. In *DocEng*.

[26] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*.

[27] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. 2016. Densecap: Fully convolutional localization networks for dense captioning. In *CVPR*.

[28] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. DVQA: Understanding data visualizations via question answering. In *CVPR*.

[29] Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. 2017. Figureqa: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300* (2017).

[30] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*.

[31] Dong-Jin Kim, Jinsoo Choi, Tae-Hyun Oh, and In So Kweon. 2019. Dense relational captioning: Triple-stream networks for relationship-based captioning. In *CVPR*.

[32] Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. 2017. A hierarchical approach for generating descriptive image paragraphs. In *CVPR*.

[33] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision* 123, 1 (2017), 32–73.

[34] Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying Task-oriented Dialogue Systems with Single Sequence-to-Sequence Architectures. In *ACL*.

[35] Mateusz Malinowski and Mario Fritz. 2014. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*.

[36] James H Martin and Daniel Jurafsky. 2009. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson/Prentice Hall Upper Saddle River.

[37] Alexander Patrick Mathews, Lexing Xie, and Xuming He. 2016. Senticap: Generating image descriptions with sentiments. In *AAAI*.

[38] Kathleen F McCoy, Sandra Carberry, Tom Roper, and Nancy Green. 2001. Towards generating textual summaries of graphs.

[39] Vibhu O. Mittal, Johanna D. Moore, Giuseppe Carenini, and Steven Roth. 1998. Describing Complex Charts in Natural Language: A Caption Generation System. *Computational Linguistics* 24, 3 (1998), 431–467.

[40] Priscilla Moraes, Gabriel Sina, Kathy McCoy, and Sandra Carberry. 2014. Generating summaries of line graphs. In *INLG*.

[41] Neha Nayak, Dilek Hakkani-Tür, Marilyn Walker, and Larry Heck. 2017. To Plan or not to Plan? Discourse Planning in Slot-Value Informed Sequence to Sequence Models for Language Generation. *Proc. Interspeech* (2017), 3339–3343.

[42] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *ACL*.

[43] Jorge Piazentin Ono, Ray (Sungsoo) Hong, Claudio T. Silva, and Juliana Freire. 2019. Why should we teach machines to read charts made for humans?. In *Human-Centered Machine Learning Perspectives Workshop, CHI*.

[44] Jorge Poco and Jeffrey Heer. 2017. Reverse-Engineering Visualizations: Recovering Visual Encodings from Chart Images. *EuroVis* (2017).

[45] Xin Qian, Eunyee Koh, Fan Du, Sungchul Kim, and Joel Chan. 2020. A Formative Study on Designing Accurate and Natural Figure Captioning Systems. In *CHI-EA*.

[46] Steven F. Roth, John Kolojejchick, Joe Mattis, and Jade Goldstein. 1994. Interactive Graphic Design Using Automatic Presentation Knowledge. In *CHI*.
[47] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017. A simple neural network module for relational reasoning. In *NIPS*.
[48] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *ACL*.
[49] Noah Siegel, Zachary Horvitz, Roie Levin, Santosh Divvala, and Ali Farhadi. 2016. FigureSeer: Parsing result-figures in research papers. In *ECCV*.
[50] Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical Phrase-Based Post-Editing. In *NAACL*.
[51] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2014. Show and Tell: A Neural Image Caption Generator. *arXiv preprint arXiv:1411.4555* (2014).
[52] W3C 2019. *A First Review of Web Accessibility.* https://www.w3.org/WAI/test-evaluate/preliminary/images
[53] Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In *EMNLP*.
[54] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*.
[55] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. 2020. Re-examining whether, why, and how human-ai interaction is uniquely difficult to design. In *CHI*.
[56] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *CVPR*.
[57] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. 2017. Boosting image captioning with attributes. In *CVPR*.
[58] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. 2018. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *NIPS*.
[59] Yue Zheng, Yali Li, and Shengjin Wang. 2019. Intention Oriented Image Captions with Guiding Objects. In *CVPR*.