Scientific Chart Summarization: Datasets and Improved Text Modeling

Hao Tan[1*†] **Chen-Tse Tsai**[2*] **Yujie He**[2*] **Mohit Bansal**[1]

[1]UNC Chapel Hill [2]Bloomberg

Abstract

Chart figures usually convey the key message in a multimodal document. Understanding charts automatically and making charts more accessible becomes indispensable in the information era. In this paper, we study the chart summarization problem in which the goal is to generate sentences that describe the salient information in a chart image. To obtain training examples, we leverage image-caption pairs in multiple scientific areas. We create a dataset of single-chart images from research papers in PubMed Central (PMC) and arXiv. Most recent vision-and-language works focus on natural images. Several challenges in structured images such as charts are under-explored. One key property of charts is that the text components (e.g., legends and axis names) carry important information. In our proposed model, we not only use a standard visual encoder but also a text encoder to encode a chart image. The visual and textual representations are connected to a large pre-trained language decoder via pre-embedding and cross-attention approaches, respectively. Experimental results show that the proposed model is significantly better than an image captioning baseline.

1 Introduction

Information graphics, such as line charts and bar charts, are essential and common components of a document. Charts are usually used for visually summarizing important information that a document intends to convey. Moreover, as shown in the study of Carberry, Elzer, and Demir (2006), information graphics in magazines and newspapers often convey messages that are not repeated in the text. Therefore, summarizing the primary message in a chart is an important step towards understanding a multimodal document. Potential applications of chart summarization include indexing information content for a search engine, making charts accessible for individuals with eyesight impairments, and simplifying information dissemination of technical visual info to a layperson.

We have seen the success of image captioning works recently, which can be viewed as generating summaries for an image. However, this research has mostly focused on natural images while other types of images (e.g., structured images shown in Fig. 2) are under-explored. On the other hand, abstractive text summarization models also have been greatly improved due to the development of neural network models. However, these models only look at the text component in a document. In this work, we focus on the less-studied yet important task of 'chart summarization', where we want to generate a salient summary for structural charts. First, to obtain a large quantity of summaries of chart images, we leverage captions in scientific articles. Unlike magazines or newspapers, in which image captions could be less descriptive, captions in scientific papers tend to be more detailed and verbose. We build a chart summarization dataset from the papers in arXiv and PubMed Central (PMC) by assuming that captions are salient summaries of chart figures. Image captions in these data sources are written by the corresponding paper's authors, and hence would be more natural in the language format. Since these articles also contain figures other than charts, we create crowdsourcing tasks to select single-chart images and collect these charts' detailed types (e.g., line chart, bar chart, etc.).

Different from the traditional captioning for natural images, there are two main challenges from the language perspective when the target images are charts: (1) Besides visual content, charts usually also contain text (e.g., legends and axis titles) which carries significant information of components in charts. (2) Charts are likely to be used in some specific domains, thus the language generation model may suffer from rare-word issues.

To address these two challenges, we first use an optical character recognition (OCR) model to detect the text boxes in the charts. An OCR embedding layer is proposed to encode these extracted texts with their position information into vectors, and these vector representations are treated as another input to the language decoder through cross-attention mechanism. Secondly, to endow the decoder with domain-specific knowledge, we use a large pre-trained language decoder instead of training it from the scratch. The chart information is connected to this pre-trained language decoder via two approaches: pre-embedding and cross attention. We empirically find that using pre-embedding for visual content and cross-attention for OCR representations gives the best results.

We apply our models on our collected datasets of two sci-

[*] Equal Contribution.

[†] Work done during an internship at Bloomberg L.P.

entific domains. We conduct both metric-based automatic evaluation and human-annotated qualitative evaluation. Experimental results show that our model with the integration of OCR and pre-trained language model significantly outperforms the baseline image captioning model. We also show the ablation studies that illustrate the effectiveness of our proposed methods.

2 Related Work

Most work on understanding chart images involves chart type classification. Savva et al. (2011) classify given chart images into 10 chart categories using an SVM classifier with visual bag-of-words and text-region features. With a similar model, Ray Choudhury and Giles (2015) proposed a binary classifier to determine whether an image is a line chart. Siegel et al. (2016) experimented with CNN-based models for classifying images they extracted from scholarly articles. In order to identify chart figures for training our summarization model, we build a binary classifier to identify common charts (e.g., line charts, bar charts, scatter plots, etc.).

There is a line of works on interpreting text components in chart images (Huang and Tan 2007; Demir, Carberry, and McCoy 2012; Chen, Cafarella, and Adar 2015; Choudhury, Wang, and Giles 2016; Kembhavi et al. 2016; Siegel et al. 2016; Kahou et al. 2018; Singh et al. 2019; Hiippala et al. 2020; Methani et al. 2020). One of the applications here is to recover visual encodings for purposes of indexing and search. For example, Poco and Heer (2017) proposed an end-to-end text analysis pipeline that identifies text elements in a chart image, determine their bounding box, and classifies their role in the chart (e.g., x-axis label, x-axis title, legend title, etc). They also proposed a CNN model that classifies the type of graphical mark (e.g., bars or lines). We simply use a general purpose OCR tool for recognizing text in chart images and focus more on the text generation model. These better text analysis models could potentially improve our model performance, which we leave for future investigation. Kahou et al. (2018) introduce FigureQA, a visual reasoning corpus of question-answer pairs over synthetic chart images. Instead of answering questions on the synthetic charts, we aim at directly summarizing real chart images.

There are some earlier works on chart summarization. Elzer et al. (2007) proposed SIGHT, a system that summarizes bar charts for visually impaired users. The system identifies one of the twelve message categories that can be conveyed by a bar chart and produces a logical form. This logic representation is then translated into natural language via templates. Demir, Carberry, and McCoy (2008) built on top of SIGHT. The proposed system first identifies an additional set of propositions that may reflect some information in a bar chart by rules. These propositions are then organized and structured by a bottom-up planner. Finally, a surface realizer is applied to produce natural language summaries.

Greenbacker, Carberry, and McCoy (2011) built a corpus of human-written English summaries of line graphs. They selected 23 line graphs and asked annotators to summarize the most important information in each graph. As this process is difficult to be scaled up, we take the captions of chart images in scientific papers to represent the summaries instead. Greenbacker et al. (2011) further used this corpus and proposed an abstractive summarization system for line charts. The system uses a Bayesian network to classify the intents of line segment, and then rules are applied to identify additional important informational propositions conveyed by the line graph. The sets of intents and prepositions are pre-defined from the study on the corpus. They left the final step of generating natural language summary from prepositions as future work. Therefore, no evaluation results were shown.

A common challenge of these earlier works is that they are limited to a fixed set of propositions and need to convert the selected propositions to natural language. Instead of using a pipeline with hand-crafted intents and propositions, we propose to leverage an end-to-end neural network, which has been shown to be powerful in generating coherent and grammatical sentences in the context of image captioning and abstractive text summarization.

Another thread of related works is (natural) image captioning, which tries to generate descriptions for natural images. Vinyals et al. (2015) first illustrate the end-to-end encoder-decoder architecture and Xu et al. (2015) extends it with attention modules. Ranzato et al. (2016) use reinforcement learning to eliminate exposure bias but requires a large amount of data to reduce the high variance. Anderson et al. (2018) take object-level information to enable fine-grained visual understanding. However, we empirically found that the detection features for natural image do not work well for charts (structural images). Previous vision-and-language pre-training, e.g., VLP (Zhou et al. 2019) and OSCAR (Li et al. 2020), use pre-trained vision-and-language model to improve image captioning but requires a large in-domain corpus and heavy pre-training.

3 Datasets Creation

We create our datasets based on image-caption pairs that appear in public scientific papers. Different from the figures in magazines or newspapers where the captions could be less descriptive, figure captions in scientific articles tend to convey the key message of figures. The assumption here is that captions written by the paper authors could represent the most salient information in the figures, therefore could serve as summaries of the corresponding figures. The overview of our datasets creation pipeline is shown in Figure 1. We consider two data sources: arXiv[1] and PMC.[2] ArXiv is a free distribution service and an open-access archive for scholarly articles in the fields such as physics, computer science, and mathematics. PMC is a free full-text archive of biomedical and life sciences journal literature at the U.S. National Institutes of Health's National Library of Medicine. We take articles in the Open Access Subset.[3] These two data sources are chosen because they both provide structural data in ad-

[1] https://arxiv.org/

[2] https://www.ncbi.nlm.nih.gov/pmc/

[3] https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/, and only use articles which have a CC BY or CC0 license so that we can release our dataset.

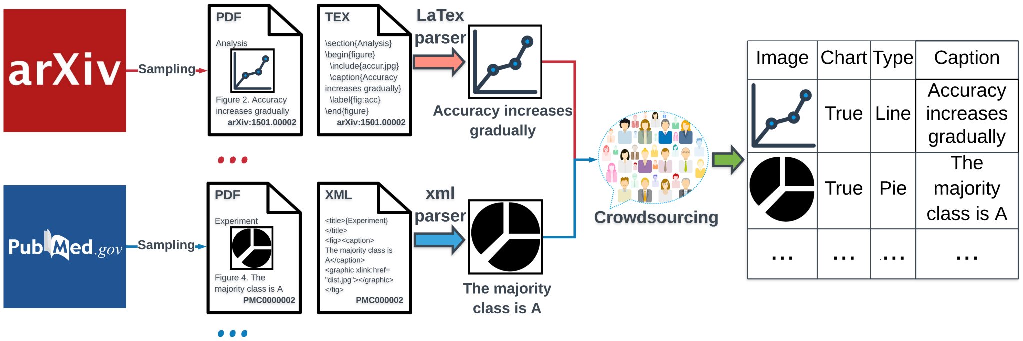

Figure 1: Pipeline of datasets creation. We first sample scientific papers from arXiv and PubMed Central, and then extract image-caption pairs by parsing the source LaTeX or XML files. Finally, crowdsourcing is applied to annotate whether an image contains a single chart and the corresponding chart type.

dition to the PDF files. That is, we can obtain image-caption pairs by parsing the LaTeX source files provided by arXiv or the XML files provided by PMC. We write our own LaTeX parser for the arXiv data, and use a public PubMed parser[4] for parsing XML information.

Although we can extract lots of image-caption pairs, most of the figures in these papers are not charts. Hence, to be able to train and evaluate the proposed chart summarization model, we need to identify which figures are charts. In this work, we focus on the common 5 chart types, including line, bar, scatter, pie, and area charts (Figure 2). Moreover, we further focus on the simplest case where images only contain a single chart. Figures with multiple charts or with any non-chart component will be considered as negative images in this work. In the following sections, we describe how do we obtain single chart and chart type annotations.

3.1 PubMed Central Data

For PMC data, we create a crowdsourcing task to annotate whether a given image contains single chart. We randomly sample 50,000 images from the papers published from 2011 to 2019. For each image, we ask annotators whether it is a single chart figure. If the answer is yes, the annotators are required to select a chart type from line, bar, scatter, pie, area, or other chart. Since this task is pretty simple, we ask two annotators to label each image in the first round. In most cases, two annotators agree on the labels. More specifically, the Fleiss' kappa scores for "whether it's a single chart" and "chart type" tasks are 0.56 and 0.73 respectively, which shows significant agreement [5].

If there is a disagreement on either single chart label or chart types, we further ask the other three annotators to perform a second round of annotation on these images. Finally, majority vote is applied to resolve conflicts among all five annotators. We note that single charts with "other" chart type are considered negative images in our experiments.

Among 50,000 images, we obtain 7,397 positive images (single chart), including 3681 line charts, 3088 bar charts, 478 scatter charts, 125 pie charts, and 25 area charts. The positive ratio of the charts is about 13%. This low ratio is because most of the figures in scientific articles are non-chart figures (e.g., model architecture diagrams). In this work, we only use chart types in analyzing model performance. That is, chart type information is not included explicitly in model training.

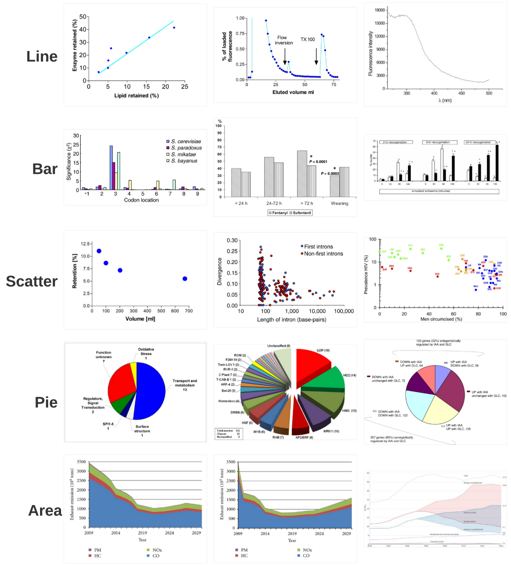

Figure 2: Example charts with the corresponding chart types from the PubMed Central dataset. The dataset we build contains the most common 5 chart types.

3.2 ArXiv Data

We also build another dataset from the arXiv data. We take papers in Computer Vision, Computation and Language, Machine Learning, Artificial Intelligence, and Neural and Evolutionary Computing fields from 2008 to 2020. Because of the copyright issue, we cannot put arXiv images on a public crowdsourcing platform. Instead, the authors went through and annotated 2000 randomly sampled figures with the same crowdsourcing interface that we use for annotating PMC data. This results in 370 single chart images. Given the

[4] https://github.com/titipata/pubmed_parser

[5] https://en.wikipedia.org/wiki/Fleiss%27_kappa

copyright issue, we only release the scripts that generate this dataset.

4 Methodology

In this section, we introduce the proposed models and training strategies for the chart summarization task. In this chart summarization task, the model needs to generate a sequence of words $\{w_i\}$ for describing the contents in a chart x. We start with introducing the basic captioning model. To enhance in-image text understanding and endow external knowledge, we incorporate an OCR encoder and a pre-trained language decoder. Lastly, we propose a simple semi-supervised learning and domain adaptation approach using a chart classifier.

4.1 Base Model

Our base model is adopted from the attentive encoder-decoder model for image captioning proposed in Xu et al. (2015). A ResNet-101 (He et al. 2016) visual feature extractor encodes the chart into a $7 \times 7 \times 2048$ dimensional feature map, where each vector in the feature map corresponds to a grid region of the image. Feature maps are then flattened to 49×2048 feature sequences $\{f_i\}$.

$$\{f_i\}_{i=1}^{49} = \mathrm{ResNet}\,(x)$$

At each decoding step t, the LSTM (Hochreiter and Schmidhuber 1997) language decoder outputs the hidden outputs h_t and cell c_t by reading the previous word w_{t-1} and states $(h_{t-1},\ c_{t-1})$. The attention module (denoted as $\mathrm{Att}_{h\to f}$) then attends to the feature sequence $\{f_i\}$ with the hidden output h_t as a query. The context $\hat{f}_t$ and the hidden vector h_t are merged into an attentive hidden vector $\hat{h}_t$ with a fully-connected layer:

$$\begin{aligned}\tilde{w}_{t-1} &= \mathrm{embedding}\,(w_{t-1})\\ h_t, c_t &= \mathrm{LSTM}\,(\tilde{w}_{t-1}, h_{t-1}, c_{t-1})\\ \hat{f}_t &= \mathrm{Att}_{h\to f}(h_t, \{f_i\})\\ \hat{h}_t &= \tanh(W_1[\hat{f}_t; h_t] + b_1)\end{aligned}$$

The probability of generating the k-th token at time step t is the softmax over a linear transformation of the attentive hidden $\hat{h}_t$. The loss $\mathcal{L}_t$ is the negative log likelihood of the ground truth token w_t^*:

$$\begin{aligned}p_t(w_{t,k}) &= \mathrm{softmax}_k\left(W_{\mathrm{w}}\,\hat{h}_t + b_{\mathrm{w}}\right)\\ \mathcal{L}_t &= -\log p_t(w_t^*)\end{aligned}$$

4.2 Text Understanding

Different from natural image captioning, the summarization of charts heavily relies on the understanding of text inside the images. However, the ResNet visual encoder (in Section 4.1) is insensitive to the text in the images (as shown in Singh et al. (2019) as well) thus we need to build a pipeline to extract the text information from the images. Specifically, we first use the Tesseract (Smith 2007) to extract a sequence of m texts $text_j$ with their positions pos_j from the image x.

$$\{(text_j, pos_j)\}_{j=1}^{m} = \mathrm{OCR}(x) \tag{1}$$

Since the characters in charts are usually in small font and sometimes blurred with the chart content, the copy mechanism (Gu et al. 2016; See, Liu, and Manning 2017) that directly brings the text into final summarization does not provide good results. We instead use the shallow text embedding layer to project the OCR text to dense vector representations that denoises the OCR detection results. We also encode the position of the OCR along with the text representation since the spatial information indicates the properties of the text (e.g., in the legend, in the title, or inside the chart):

$$g_j = \mathrm{Emb}_{\mathrm{TEXT}}(text_j) + W_{\mathrm{POS}}\,pos_j \tag{2}$$

These OCR representations are treated as another view of the charts and the language decoder simultaneously attends to the OCR information $\{g_i\}$ and visual image features $\{f_j\}$. The final hidden output $\tilde{h}_t$ is calculated based on the concatenation of the visually attended vector $\tilde{f}$, the OCR attended vector $\tilde{g}$, and the hidden state h_t.

$$\tilde{f} = \mathrm{Att}_{h\to f}(h_t, \{f_i\}) \tag{3}$$

$$\tilde{g} = \mathrm{Att}_{h\to g}(h_t, \{g_j\}) \tag{4}$$

$$\tilde{h}_t = \tanh(W_2[\tilde{f}, \tilde{g}, h_t] + b_2) \tag{5}$$

We next replace the original attentive hidden $\hat{h}_t$ with this OCR-enhanced hidden output $\tilde{h}_t$ (in Sec. 4.1) in succeeding decoding steps.

4.3 Pre-trained Language Decoder

When summarizing charts in news or scientific papers, a faithful description of the chart contents also relies on external knowledge, and hence a pre-trained language decoder might help the generation. As shown in Figure 3, we illustrate our model which integrates a pre-trained language decoder GPT-2 (Radford et al. 2019).[6] As described in the previous section, we have two image encoders (i.e., ResNet encoder and OCR text encoder) to process the image content and image text respectively. The ResNet encoder maps the features into a squared feature map (the purple vector blocks in Figure 3) where each vector corresponds to a part of image content. We will view this feature map as a sequence of vectors (as in Eq. 1) in the following procedures. The OCR encoder (Eq. 4.2) maps the chart into a sequence of recognized words and their positions on the chart. The OCR embedding layer (Eq. 2) adds the word embedding and the position encoding into one vector for each OCR entry (the yellow vectors in Figure 3).

In order to connect these visual and textual information from the image to the language decoder, we adopt two ways: appending pre-embeddings and adding cross-attention layers. The pre-embedding approach is to concatenate the sequence of visual vectors before the word embeddings thus the language decoder will take this concatenation as input (e.g., the concatenation of red blocks and blue blocks in Figure 3). The cross-attention approach adds cross-attention

[6] The method could also be applied to other pre-trained language decoders such as XLNet (Yang et al. 2019), T5 (Raffel et al. 2019), and BART (Lewis et al. 2020).

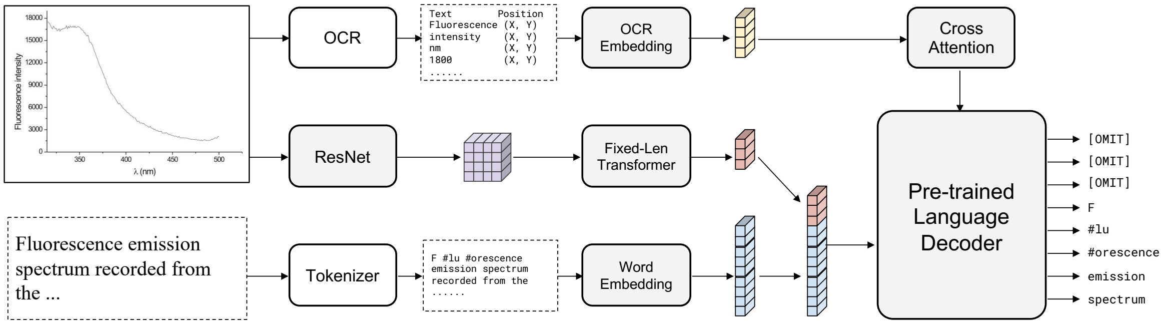

Figure 3: Illustration of the proposed chart summarization model. We have two branches of image encoding: (1) the visual branch via the ResNet and fixed-length transformer (2) the text branch via the OCR system and the OCR embedding layer. The output of these two branches are then fused into the pre-trained language decoder by pre-embedding (concatenation) and cross-attention layer, respectively. The grey boxes are neural networks.

layers (Vaswani et al. 2017) inside the language decoder to fuse visual information. The cross-attention layers contain residual short-cut connections thus the decoder still benefits from the pre-trained weights with these additional layers.

As shown in Figure 3, we use the pre-embedding approach for the features from the visual image content (i.e., from the ResNet encoder) and use the cross-attention layers for the OCR texts. The idea of this specific design is that the generation would be led by the image content and will use the OCR information to generate concrete words. We empirically find that it is the best combination to fuse information into the language decoder, and we show the comparison in Section 6.2. In detail, the length of the ResNet feature map is 49 and the order of the features is not aligned with the positional encoding in the pre-trained language decoder. We thus do not directly append it before the word embedding but use a fixed-length transformer to map it to a sequence of 10 vectors (the red blocks in Figure 3; we only draw 3 vectors for simplicity). The fixed-length transformer is built by transformer decoder layers (Vaswani et al. 2017) with only positional embedding (without word embedding). We use only 1 layer in our experiments.

4.4 Semi-Supervised Learning and Domain Adaptation

Although we can extract abundant image-caption pairs, most figures in scientific articles do not contain a chart as we discussed in Section 3. If we want to reserve enough human-annotated examples for the metric-based evaluation purpose, that leaves very little data for training, especially for the arXiv domain in which we only have hundreds of single-chart images. Therefore, we leverage semi-supervised learning techniques to take advantage of large unannotated data and use domain adaption to transfer to other datasets. Both of these two methods rely on a chart classifier that we will introduce first.

Chart Classifier. The key component in getting more training examples is a classifier that can identify single-chart images. We take the ResNet (He et al. 2016) as the visual backbone and use a binary linear classifier after the mean-pooled features. Instead of freezing the backbone model as in the previous works (Xu et al. 2015), we fine-tune the classifier with a small learning rate, 10^{-4}. We find that this standard classifier reaches good results (see Appendix for details).

Semi-Supervised Learning. In the semi-supervised learning setup, we have labeled data (Section 3) and we want to improve the performance from the unlabeled data. The unlabeled data contains both charts and non-chart images (e.g., model figures in scientific publications and natural images in news). Including these non-chart images in training data will introduce noise and thus lead to an increment in training time. To provide clean data in semi-supervised learning, we filter the unlabeled data with our chart classifier and train the summarization model based on the filtered data. In this way, we increase the amount of data and the coverage of topics.

Domain Adaptation. Different from semi-supervised learning, domain adaptation focuses on transferring the labeled dataset into another domain. Naïve transferring without training on the target domain would under-fit the target distribution and we empirically show its ineffectiveness in Appendix. To solve this issue, we use a similar approach to the semi-supervised learning that trains the proposed summarization model on the dataset created by the chart classifier. More specifically, since we have much less labeled charts in the arXiv domain, we treat it as the target domain whereas PMC data is the source domain. We train the chart classifier on the PMC data, and apply it on the images from arXiv papers to obtain large amount of single-chart images.

5 Results

In this section, we evaluate our proposed methods on our collected datasets of two domains: PMC and arXiv. We start with describing the experiment setups and show results with both automatic metric-based evaluation and human evaluation.

5.1 Experimental Setup

Data Setup. The supervised learning setup is conducted on our annotated PMC dataset. We randomly sample 1,000 charts as the test set and split the remaining charts into training (5,819) and validation (646) sets with a ratio of 9:1.

In order to increase the number of training examples, we apply the proposed semi-supervised learning technique (Section 4.4). The single-chart classifier is based on the

	PMC (Supervised)				PMC (Semi-Supervised)				arXiv (Domain Adaptation)			
	BLEU	ROUGE-L	METEOR	CIDEr	BLEU	ROUGE-L	METEOR	CIDEr	BLEU	ROUGE-L	METEOR	CIDEr
Base Model	1.66	11.35	2.77	2.76	2.09	11.05	2.91	4.49	3.55	14.10	3.79	8.99
+ OCR	1.97	**11.77**	3.09	**6.00**	2.53	11.95	3.50	7.98	4.78	**15.88**	4.68	15.88
+ GPT-2	**3.19**	11.66	**3.68**	1.57	**4.47**	**12.46**	**4.32**	**10.30**	**5.89**	14.32	**4.92**	**32.34**

Table 1: Results on the PubMed Central (PMC) and arXiv datasets. Supervised: training images are human-labeled single chart images. Semi-Supervised: training images also include the positive images from the proposed chart classifier. Domain Adaptation: the chart classifier trained on the PMC domain is applied on arXiv domain to obtain training data for the summarization model. The best results are marked in bold.

ResNet-101 model and is fine-tuned on our datasets. We use the 50,000 human-labeled images (7,465 positives) from PMC data to build this binary classifier. After the model converges on the training set, we calibrate the classifier to optimize the recall with an precision over 99% on the validation set. Since we have lots of images, we can afford a lower recall for high-quality positive examples. We then use this classifier to filter the unlabeled images in the PMC data to augment the training set. More specifically, besides the 50,000 images we used in the crowdsourcing task, there are 137,928 remaining articles in our PMC collection from the year of 2011 to 2019. After applying the chart classifier, we obtain 13,637 single chart images which could serve as additional training examples for the summarization model.

For domain adaptation, we take charts and captions from arXiv as the target domain. As described in Section 3, we have manually annotated 370 single-chart images in this domain, which are served as the test set. We use the same chart classifier in the previous semi-supervised learning setup to annotate 140,000 arXiv images. This results in 22,044 positive examples. We split this 22,044 examples into training data (19,840) and validation data (2,204) with a ratio of 9:1.

Model Setup. For the base model, we use a ResNet-101 model from the Torchvision (Marcel and Rodriguez 2010) library[7]. We resize the image into 224 × 224 and the backbone model maps it to a 7 × 7 × 2048 vectors. We sort the OCR-extracted texts by their confidence and only keep the top 20 texts for post-processing. Since we want the image position to be related to the OCR position. We do not apply random resize and cropping but directly resize the chart into 224 × 224. For the pre-trained GPT-2 (Radford et al. 2019) model, we downloaded the small GPT-2 model from Hugging Face's Transformer (Wolf et al. 2020). The GPT-2 implementation has support of cross-attention layers as in Vaswani et al. (2017) and we use it to attention to the OCR features. For the fixed-length transformer, we use 1 layer with the same architecture as the GPT-2 model but do not apply the causal attention mask. More implementation and hyperparameter details can be found in Appendix.

	Baseline Better	Final Model Better	Equally Good	Equally Bad Bad
PMC	20	70	3	7
arXiv	37	50	2	11

Table 2: Human study on the results with 100 pairwise comparisons.

5.2 Metric-based Evaluation

In order to conduct efficient evaluation, we take the automatic language metrics to evaluate our model. We report the BLEU (Papineni et al. 2002), ROUGE-L (Lin 2004), METEOR (Banerjee and Lavie 2005), and CIDEr (Vedantam, Lawrence Zitnick, and Parikh 2015) as in previous image captioning papers. As shown in Table 1, we compare our proposed models (in Section 4.2 and Section 4.3) with the baseline captioning model (in Section 4.1) on both PMC and arXiv datasets. The model with OCR text encoder is strictly better than the baseline captioning model for every metrics, which indicates that the in-chart text understanding is very important for generating good summarization for scientific charts. The integration of the pre-trained language model (GPT-2) further enhances the performance over the OCR encoder results. The pre-trained decoder shows more improvement on the semi-supervised setup since the model needs enough data to learn the weights in the fixed-length transformer and the cross-attention modules, which bridge the vision encoder and the language decoder.

Note that the CIDEr score of the +GPT-2 model is lower than the +OCR model on the PMC dataset under the supervised setup. We find that this is due to the size of data. The smaller size of the PMC data makes the learned model have a stronger bias towards the original GPT-2 generation. Namely, although the model would generate more fluent sentences (reflected on the high BLEU score), it is biased towards the GPT-2 prior by leveraging mostly common words. This bias is captured by the CIDEr metric's over-weighting protocol. However, under the semi-supervised setting, the CIDEr score is higher with GPT-2 because of the adequate amount of data. This also demonstrates the usefulness of the proposed semi-supervised approach.

5.3 Human Evaluation

In order to get a faithful evaluation, we conduct a human evaluation on 100 randomly sampled examples for PMC and arXiv. The human evaluation is conducted by the authors and their colleagues (4 in total) since this task requires a certain expert knowledge. We use both base captioning model and our final model (with OCR encoder and GPT-2 decoder)[8]

[7] https://pytorch.org/docs/stable/torchvision/models.html

[8] The PMC model is with the semi-supervised setup.

	BLEU	ROUGE-L	METEOR	CIDEr
All	4.47	12.46	4.32	10.30
Line Chart	4.44	12.70	4.28	10.18
Bar Chart	4.77	12.30	4.71	7.14
Scatter Chart	5.96	16.63	5.39	40.78

Table 3: Results regarding different types of charts.

to generate two summaries. Each image with the generated summaries from the two models is annotated by all 4 annotators. We randomly shuffle the order of these two summaries and only show the A/B labels to the human annotators. The human annotators is asked to choose one from the four options: "Both Good", "Both Bad", "A wins", and "B wins". As shown in Table 2, our proposed model significantly outperforms the baseline model for both datasets. Moreover, we find that our annotators have a high agreement on which generated sentence is better since this scientific summarization is mostly about facts and salience.

6 Analysis

In this section, we provide the fine-grained analysis to illustrate the effectiveness of each component in the proposed pipeline. We first demonstrate the results for different chart types and cross-domain evaluation in Section 6.1. In Section 6.2, we empirically show the advantage of our pre-embedding and cross-attention combination.

6.1 Different Chart Categories

During our data collection, we also let the annotators to select the type of the chart (Figure 2). In this paper, we aim for a general chart summarization model that does not rely on the details of each chart type. We here analyze the performance of the proposed model on each chart category with our final model trained on PMC (Semi-Supervised). In Table 3, we show the results of the most common three chart types (i.e., "Line", "Bar", "Scatter") that have sufficient amount of data (513 for Line, 400 for Bar, and 57 for Scatter) to support automatic metric-based evaluation. Although the line charts contribute the most to the training and test data, the BLEU score is the lowest compared to the results of bar charts and scatter charts. The reason might be that the image features produced by convolutional neural networks (CNN) are insensitive to the properties (e.g., trending, crossings) of the curved lines. At the same time, the CNN could capture the local intensity of points thus show higher results for scatter chart. According to this observation, we think that using visual encoder that are specifically designed for understanding the curved lines in chart might be a promising future direction.

6.2 Pre-Embeddings and Cross-Attention Layers

In Section 4.3, we discuss two ways to connect the visual information to the language decoder: the pre-embedding approach and the additional cross-attention layers. In Table 4, we show the results of different combinations on PMC (semi-supervised) dataset. "Img" and "OCR" indicates using the image output and OCR representations as the input to the pre-embedding approach and the cross-attention layers. "None" means that we do not use input and thus excludes the parameters. "Concat" means that we concatenate the output of image and OCR representations together and use it as the input. We can see that the our approach (Img for Pre-Embed and OCR for Cross-Att) is comparable to its reverse (OCR for Pre-Embed and Img for Cross-Att) and is much better than other alternatives.

Pre-Embed	Cross-Att	BLEU	ROUGE-L	METEOR	CIDEr
None	None	1.91	10.59	3.01	0.52
Concat	None	2.88	11.92	3.79	4.78
None	Concat	3.64	12.07	3.69	2.91
Img	OCR	4.47	12.46	4.32	10.30
OCR	Img	4.46	12.12	4.08	11.18
Concat	Concat	3.61	12.18	3.76	2.79

Table 4: Comparison of different approaches of connecting the image content and the language decoder.

6.3 Chart Classification Performance

In both the semi-supervised learning and domain adaption setup, we use a classifier to identify single-chart images from lots of automatically extracted image-caption pairs. Since the images filtered by the classifier will be further used as data augmentation, we take the F_1 score as the main metric to balance the precision and recall. We start with the frozen ResNet-101 (He et al. 2016) features with an additional linear classifier. This setup achieves 90% F_1 score. After fine-tuning the backbone model on our data, the model achieves an F_1 score of 94.9%. We also tried adding other neural modules (e.g., attentive modules and detection branches) and enhanced visual backbones but we do not observer a significant result improvement on the test set.

When we use this classifier in the semi-supervised and domain adaptation setups, we calibrate the classification threshold to maintain a precision over 99% since we have lots of unannotated images. Under this precision level, we achieve a recall of 59.8% and precision of 99.2%. We kept the same classification threshold and test it on our annotated arXiv test split. The precision and recall are 93.4% and 65.7%, respectively.

7 Conclusions

In this paper, we propose datasets and models for summarizing scientific charts, a specific type of structured images. We construct datasets from PMC and arXiv by leveraging crowdsourcing and the figure captions in the papers. To enable better understanding text components in charts and to endow the model with external knowledge, we propose to use an OCR encoder and a pre-trained language decoder on top of a standard image captioning model. In our experiments, we show the effectiveness of our models in terms of both automatic evaluation metrics and human evaluation.

8 Acknowledgements

The authors thank Bloomberg's AI Engineering team, especially Alakananda Vempala, Ketevan Tsereteli, and Anju Kambadur for helpful feedback and directions. Additional thanks to the anonymous reviewers for their insights. Hao Tan acknowledges support from Bloomberg's Data Science Ph.D. Fellowship.

References

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6077–6086.

Banerjee, S.; and Lavie, A. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 65–72.

Carberry, S.; Elzer, S.; and Demir, S. 2006. Information graphics: an untapped resource for digital libraries. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 581–588.

Chen, Z.; Cafarella, M.; and Adar, E. 2015. Diagramflyer: A search engine for data-driven diagrams. In *Proceedings of the 24th International Conference on World Wide Web*, 183–186.

Choudhury, S. R.; Wang, S.; and Giles, C. L. 2016. Scalable algorithms for scholarly figure mining and semantics. In *Proceedings of the International Workshop on Semantic Big Data*, 1–6.

Demir, S.; Carberry, S.; and McCoy, K. 2008. Generating Textual Summaries of Bar Charts. In *Proceedings of the Fifth International Natural Language Generation Conference*, 7–15. Salt Fork, Ohio, USA: Association for Computational Linguistics.

Demir, S.; Carberry, S.; and McCoy, K. F. 2012. Summarizing information graphics textually. *Computational Linguistics*, 38(3): 527–574.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*.

Elzer, S.; Schwartz, E.; Carberry, S.; Chester, D.; Demir, S.; and Wu, P. 2007. A Browser Extension for Providing Visually Impaired Users Access to the Content of Bar Charts on the Web. In *WEBIST (2)*, 59–66. Citeseer.

Greenbacker, C.; Carberry, S.; and McCoy, K. 2011. A Corpus of Human-written Summaries of Line Graphs. In *Proceedings of the UCNLG+Eval: Language Generation and Evaluation Workshop*, 23–27. Edinburgh, Scotland: Association for Computational Linguistics.

Greenbacker, C.; Wu, P.; Carberry, S.; McCoy, K.; and Elzer, S. 2011. Abstractive Summarization of Line Graphs from Popular Media. In *Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*, 41–48. Portland, Oregon: Association for Computational Linguistics.

Gu, J.; Lu, Z.; Li, H.; and Li, V. O. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1631–1640.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hiippala, T.; Alikhani, M.; Haverinen, J.; Kalliokoski, T.; Logacheva, E.; Orekhova, S.; Tuomainen, A.; Stone, M.; and Bateman, J. A. 2020. AI2D-RST: A multimodal corpus of 1000 primary school science diagrams. *Language Resources and Evaluation*, 1–28.

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

Huang, W.; and Tan, C. L. 2007. A system for understanding imaged infographics and its applications. In *Proceedings of the 2007 ACM symposium on Document engineering*, 9–18.

Kahou, S. E.; Atkinson, A.; Michalski, V.; Kádár, Á.; Trischler, A.; and Bengio, Y. 2018. FigureQA: An Annotated Figure Dataset for Visual Reasoning. In *ICLR Workshop*.

Kembhavi, A.; Salvato, M.; Kolve, E.; Seo, M.; Hajishirzi, H.; and Farhadi, A. 2016. A diagram is worth a dozen images. In *European Conference on Computer Vision*, 235–251. Springer.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.

Li, X.; Yin, X.; Li, C.; Zhang, P.; Hu, X.; Zhang, L.; Wang, L.; Hu, H.; Dong, L.; Wei, F.; et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, 121–137. Springer.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74–81.

Marcel, S.; and Rodriguez, Y. 2010. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM international conference on Multimedia*, 1485–1488.

Methani, N.; Ganguly, P.; Khapra, M. M.; and Kumar, P. 2020. Plotqa: Reasoning over scientific plots. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1527–1536.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, 311–318. Association for Computational Linguistics.

Poco, J.; and Heer, J. 2017. Reverse-engineering visualizations: Recovering visual encodings from chart images. In *Computer Graphics Forum*, volume 36, 353–363. Wiley Online Library.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

Ranzato, M.; Chopra, S.; Auli, M.; and Zaremba, W. 2016. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*.

Ray Choudhury, S.; and Giles, C. L. 2015. An architecture for information extraction from figures in digital libraries. In *Proceedings of the 24th International Conference on World Wide Web*, 667–672.

Savva, M.; Kong, N.; Chhajta, A.; Fei-Fei, L.; Agrawala, M.; and Heer, J. 2011. Revision: Automated classification, analysis and redesign of chart images. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 393–402.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1073–1083.

Siegel, N.; Horvitz, Z.; Levin, R.; Divvala, S.; and Farhadi, A. 2016. FigureSeer: Parsing result-figures in research papers. In *European Conference on Computer Vision*, 664–680. Springer.

Singh, A.; Natarajan, V.; Shah, M.; Jiang, Y.; Chen, X.; Batra, D.; Parikh, D.; and Rohrbach, M. 2019. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8317–8326.

Smith, R. 2007. An overview of the Tesseract OCR engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, 629–633. IEEE.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Vedantam, R.; Lawrence Zitnick, C.; and Parikh, D. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4566–4575.

Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3156–3164.

Wolf, T.; Chaumond, J.; Debut, L.; Sanh, V.; Delangue, C.; Moi, A.; Cistac, P.; Funtowicz, M.; Davison, J.; Shleifer, S.; et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, 2048–2057.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, 5753–5763.

Zhou, L.; Palangi, H.; Zhang, L.; Hu, H.; Corso, J. J.; and Gao, J. 2019. Unified Vision-Language Pre-Training for Image Captioning and VQA. In *AAAI*.

9 Implementation Details

The supervised learning setup is conducted on our annotated English PMC dataset in Sec. 3. We kept 1,000 charts in the test set and split the the remaining charts into training(5,819)/validation(646) with a ratio of 9:1. We train our model on the training set and tune the hyperparamters on the validation set. The test set is only used to report results. We train for 200 epochs on this small dataset. All our code are written in PyTorch and all experiments converge in 4 5 hours on 1 Titan V GPU.

For the base model, we use a ResNet-101 model from the Torchvision (Marcel and Rodriguez 2010) library [9]. We resize the image into 224 x 224 and the backbone model maps it to a 7 x 7 x 2048 vectors. We use 512 dimensions for the LSTM and 256 dimensions for the word embedding. The attentive hidden states has the same size as the hidden states (512 dimensions). We use an Adam (Kingma and Ba 2015) with a fixed learning rate of 10^{-4}. The batch size is 64.

For the OCR model, we sort the ocr texts by their confidence and remove the empty text. We kept the top 20 ocr texts for post-processing. We use 512 dimensions for the OCR feature representations (yellow blocks in Fig. 3). Since we want the image position to be related to the OCR position. We did not do random resize and cropping but directly resize the chart into 224 x 224.

For the pre-trained GPT-2 (Radford et al. 2019) model, we downloaded the small GPT-2 model (124M parameters) from Hugging Face's Transformer (Wolf et al. 2020) [10]. The GPT-2 implementation has support of cross-attention layers as in Vaswani et al. (2017) and we use it to attention to the OCR features. For the fixed-length transformer, we use 1 layer with the same architecture as the GPT-2 model but do not apply the causal attention mask. We use an Adam (Kingma and Ba 2015) with weight decay of 0.01 following the practice in Devlin et al. (2019). We do not use weight decay for the layer normalization layer and bias. We use a linear warmup with a peak learning rate at 10^{-4}. The first 5% steps are warmup steps. The batch size is 64.

[9] https://pytorch.org/docs/stable/torchvision/models.html

[10] https://github.com/huggingface/transformers

	PMC				arXiv			
	BLEU	ROUGE-L	METEOR	CIDEr	BLEU	ROUGE-L	METEOR	CIDEr
PMC	4.47	12.46	4.32	10.30	0.06	8.19	1.93	0.63
arXiv	0.22	10.11	3.25	1.43	5.89	14.32	4.92	32.34

Table 5: The transferability of our captioning model across different domains. The columns indicate the training dataset while the rows indicate the testing dataset. The PMC training data is augmented with filtered charts (in Sec. 4.4) and the arXiv training data is built by the chart classifier. All test data are human-annotated.

In order to increase the number of training examples, we apply the proposed semi-supervised learning technique. The single-chart classifier is based on the ResNet-101 model and is fine-tuned on our datasets. We use the 50,000 human-labeled images (7,465 positives) from PMC data to build this classifier. The training, validation, and test sets have 5,819, 646, and 1,000 data point, respectively. The data split is the same as the above supervised learning setup. After the model converges on the training set, we calibrate the classifier to optimize the recall with an precision over 99% on the validation set. Since we have lots of images, we can afford a lower recall for high-quality positive examples.

We then use this classifier to filter the unlabeled images in the PMC data to augment the training set. More specifically, besides the 50,000 images we used in the crowdsourcing task, there are 137,928 remaining articles in our PMC collection from the year of 2011 to 2019. After applying the chart classifier, we obtain 13,637 single chart images which could serve as additional training examples for the summarization model. The hyper-parameters of the summarization model is the same as the ones used in the supervised setup. For the models trained on this dataset, we use a max sequence of 80 and train for 100 epochs. The other hyperparameters are same as the small supervised PMC data for each model.

For domain adaptation, we take charts and captions from English arXiv as the target domain. As described in the dataset section, we have manually annotated 370 single-chart images in this domain, which are served as the test set. We use the same chart classifier in the previous semi-supervised learning setup to annotate 140,000 arXiv images. This results in 22,044 positive examples. We split this 22,044 examples into training data (19,840) and validation data (2,204) with a ratio of 9:1. The summarization model is trained on the training data, tuned on the validation data, and finally evaluated on the manually-annotated test set. For the models trained on this dataset, we use a max sequence of 40 since the captions in arXiv are much shorter. Since we halve the max sequence, we train for 200 epochs thus roughly keep the same computational resources for both datasets.

10 Details of Data Collection

The crowdsourcing task is conducted on Appen[11]. There are 2263 distinct annotators from 50 countries. Since the task is to classify image types, it doesn't require native English speakers. The top 5 countries are Venezuela (53%), USA (23%), Egypt (8%), Colombia (2%), and Peru (1.4%). We paid one cent per judgement (image). For the first round of annotation tasks, the Fleiss' kappa scores for "whether it's a single chart" and "chart type" tasks are 0.56 and 0.73 respectively, which shows pretty significant agreement.

11 Additional Analysis

11.1 Cross-Domain Transferability

To illustrate the need of domain adaption led by the chart classifier (in Sec. 4.4), we show the low cross-domain transferability of models in this section. Each row in Table 5 indicates the results of our final model trained on the designated dataset while each line in the Table indicate the evaluation results on the test set. The model does not transfer well between different domains, probably because the different figuring and captioning conventions from different communities. The different topics also introduce diverging vocabularies.

12 Ethical Considerations

The technique developed in this paper would help automatic summarize news, articles, and publications where charts are involved in. It would also help visually impaired people to understand the content of the charts. It would fail in cases when the OCR detector miss the key information of the charts and would lead to unfaithful summarization of the chart. Since we use a pre-trained language decoder in our final model, the generated summarization might be biased towards the pre-training domain of the language decoder. Regrading the dataset collection, we have resolved all legal and licenses issue for the PMC dataset before showing them to annotators. More specifically, we only use articles with CC BY or CC0 licenses from the Open Access Subset of PMC data. For arXiv data, we annotate a small test set by the authors. We will not directly release the arXiv images given the license constraint. Instead, we plan to release scripts that construct the dataset we built. The users will have to download arXiv packages by themselves, and then run our scripts to extract image-caption pairs and assign the labels we annotated.

[11] client.appen.com