Summarizing Charts and Graphs with Context

Stanford CS224N Custom Project

Akankshita Dash
Department of Computer Science
Stanford University
akdash@stanford.edu

Nandita Naik
Department of Computer Science
Stanford University
nanditan@stanford.edu

Abstract

Generating natural language summaries of charts and graphs is crucial to ensure accessibility for blind and low vision (BLV) users. Most current approaches to summarization do not consider the overarching context in which the chart or graph appears. However, research indicates that people who are BLV often prefer image descriptions that take context into account. To close this gap, we present a set of deep learning models that generate summaries for charts and graphs. We compare the results of models without context to models that incorporate context, and find that when we train our model on a variety of contexts, our model performs better on all metrics. Furthermore, we explore which types of context enhance summary generation by examining how the model generalizes when trained on a specific type of context. Our results underscore the importance of considering the communicative purpose of images when generating summaries.

1 Key Information to include

- Mentor: Yuan Gao
- External Collaborators (if you have any): Elisa Kreiss
- Sharing project: Yes, research for Stanford NLP Group
- Late days used: 3, shared between Akankshita (4) and Nandita (3)

2 Introduction

Charts and graphs aim to present quantitative data in a visually compelling way. Given the abundance of charts and graphs, it is important to ensure that people have equal access to the content of these images, particularly people who are blind or low-vision (BLV). Previous work shows that human-authored alt text for charts and graphs is often absent or uninformative, and is insufficient to address the needs of BLV people [1]. Therefore, we focus on the problem of automatically generating natural language summaries for charts and graphs.

While generating image descriptions is an active subfield in bridging image-text relations, summarizing charts and graphs poses an additional set of challenges [2, 3]. Charts and graphs often contain textual data in their labels, and generating chart summaries requires the model to parse text within the image, which is challenging. Studies show that visually inferring key insights from data is a task that is challenging even for humans, because people need to visually compare and contrast between many items in the data [4]. In addition, figure captions generally tend to be longer than other types of image descriptions, so the possibility of a mistake increases [5].

We hypothesize that taking into account the surrounding **context** of the graph will help create more informative summaries. Prior work in the field of generating image descriptions shows that what BLV users want in image descriptions vary with the context in the image appears [6, 7, 8]. We extend this hypothesis to charts and graphs. We conducted experiments to answer two main research questions: (1) Does context improve chart summaries when we train a model on different types of

context (e.g. market graphs, scientific figures, etc)? (2) How does a model trained on one type of context generalize to other contexts?

To evaluate our hypothesis, we trained context and no-context models. Our context models take in as input an image and a context, and output a summary of the chart. Our no-context models take in the image as input, and output a summary of the chart.

We used datasets with naturally occurring charts and graphs in varying contexts: Pew [3], a public policy research center, Statista [3], an online platform for market and consumer data, Concadia [8], which is a subset of images of charts and graphs sourced from Wikipedia, and Alt-Text HCI [1], a dataset consisting of scientific figures in human-computer interaction (HCI) papers.

For our first experiment, we concatenated all the datasets together and found that incorporating context into our models improves the scores on all metrics, and creates more informative summaries. In our second experiment, we explored if a model trained on a single type of context (Statista) generalizes to unseen datasets, and find that although the evaluation metrics of this model are slightly lower the previous one, it is not significantly inferior, and we can assert that training on Statista generalizes effectively to the other datasets.

Our key contributions include: (1) defining summaries as an effective method for communicating insights from charts and graphs, (2) training and evaluating existing context and no-context models for our purposes, and changing their attention mechanism, and (3) finetuning BLIP [9] to work with our dataset.

3 Related Work

3.1 Summarizing Charts and Graphs

Xu et al (2015) [10] formulate the problem of image captioning as machine translation, since it involves "translating" an image into sentences. The encoder extracts features from images, and the decoder translates the features into natural language sentences. Drawing from this paper, we train similar encoder-decoder models (ResNet-LSTM, and DenseNet-LSTM) in our own work.

Kantharaj et al (2022) [3] released a large-scale benchmark and dataset for the problem of chart summarization. The dataset consists of $44,096$ charts from Pew and Statista with a diverse range of topics and chart types. The researchers also benchmarked a variety of state-of-the-art models on this dataset, which we use for our project. They address two versions of the problem: the version where the underlying data table is available, and the other when it isn't. However, their models do not consider the underlying context in which the graph appears, which we hypothesize will help the model generate more informative chart summaries.

Chintalapati et al (2022) [1] conduct a study of the alt text for charts and graphs in papers submitted to ACM ACCESS and CHI, premier accessibility conferences. They find that while most alt text about graphs only contains information about basic low-level visual details, such as graph type and labels of axes [1], only 50% of the alt texts of figures discuss outliers, and 31% discuss trends. They also released a dataset of these real-world graphs scraped from HCI publications, which we included in our concatenated dataset.

Lundgard and Satyanarayan [11] describe a framework for understanding semantic content in descriptions of charts and graphs. They characterize four levels of semantic content - (1) Identify low-level visualization details (e.g. the type of figure, axis labels, etc.), (2) Report statistical concepts and relations (e.g. outliers, correlations, etc.) (3) Explain high-level patterns in the data (e.g. trends, patterns, etc.) and (4) Articulate domain-specific insights or the societal context for the data.

The authors conducted studies with BLV people, which showed that users gained the most information from textual chart and graphs descriptions in semantic levels 1 to 3 [11]. In prior research with the Stanford NLP Group, we evaluated the datasets collected using four levels of semantic content criteria and found that the descriptions for Alt-text HCI, Statista, and Pew contain mostly Level 2 data $(50-67\%)$, with varying percentages of Level 1 and Level 3 data $(6-36\%)$ and negligible Level 4 data. These findings indicate a need for a comprehensive dataset that has sufficient levels of semantic content (1-3) to create more accurate chart-to-text generation models, and show that there is still more work in order to make chart summaries more informative and relevant. Despite these limitations, we note that a concatenation of these datasets is appropriate as it is sufficiently large

(44, 674 datapoints), has fair levels of Level 2 data, and can function as a starting baseline for our task.

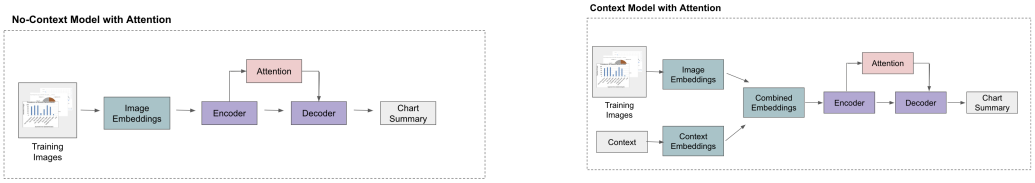
3.2 Incorporating Context in Image Descriptions

Traditionally, generating image descriptions takes a one-size-fits-all approach, using a single description of an image across many different contexts. Stangl et al (2021) [7] propose generating image descriptions that vary based on context. To support these claims, they conducted a study with 28 BLV people, and found that the information that people wanted in an image description varied depending on the scenario they were given [7]. For instance, if an image appeared on an online shopping website, participants wanted to know more about the clothes. If we extend this idea to charts and graphs, we can see, for example, how a chart or graph appearing in an online textbook might require different types of description detail as compared to a chart or graph appearing on a sports analytics website.

Kreiss et al (2022) demonstrated that augmenting image-to-text models with context generates purposeful captions and descriptions. To measure the impact of context on the quality of text generation from images [8], the researchers collected data from Wikipedia and assembled a corpus of 96,918 images with corresponding descriptions, captions, and surrounding context. They integrated the context with the image by concatenating the image embeddings with the context embeddings from BERT. We adapted their models to work with our dataset of charts and graphs.

Kreiss et al (2022) [8] also draw an important distinction between the phrase caption and description. A **caption** is intended to appear alongside the image to add supplementary information, while a **description** is intended to replace the image. The field of generating descriptions for charts and graphs suffers from a paucity of real-world data [1]. Due to this data scarcity, we couldn't train a model to generate only captions or only descriptions, and so we had to concatenate together both caption datasets and description datasets. To avoid confusion, we use the term "**summary**" broadly to encompass both descriptions and captions, and to cover all natural language generation of charts and graphs.

4 Approach



(a) No context model

(b) Context model

Figure 1: Model architecture

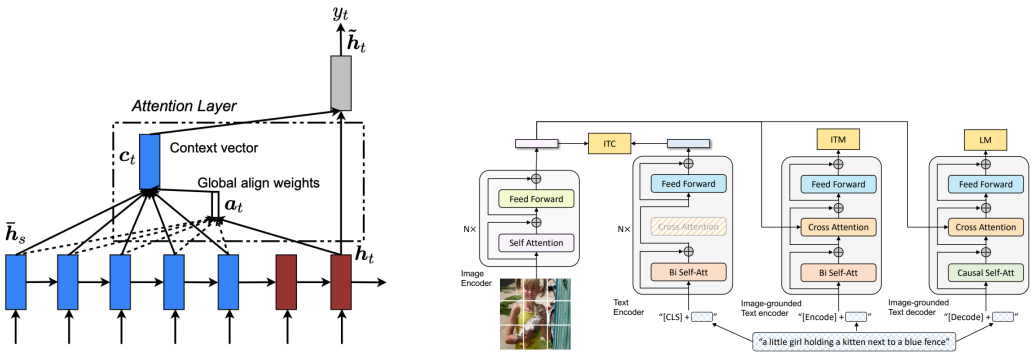
For the first experiment, we trained and evaluated two models for our chart-summary generation - DenseNet-LSTM and ResNet-LSTM [10], which are both based on encoder-decoder architecture with attention. We trained two models for each: a model without context and a model with context. For the no-context models, we use all-ones vectors in place of the context embeddings.

For our second experiment, we trained DenseNet-LSTM and fine-tuned the Bidirectional Language Model Pretraining (BLIP)[9] model. BLIP [9] is a vision-language model from Salesforce that can be used for image captioning, and makes use of a multimodal encoder-decoder architecture. We considered other pretrained models, such as CLIP [12] and BLIP-2 [13], but based on the work of Mao et al. [14], we found that CLIP doesn't generalize well to contextual data, and BLIP-2 required distributed computing for training, which was not feasible with our limited computing resources (in both cost and memory).

4.1 Methods

4.1.1 ResNet-LSTM/DenseNet-LSTM

ResNet-LSTM and DenseNet-LSTM are both encoder-decoder architectures that combine either a ResNet or a DenseNet-based image encoder (from which we remove the last classification layer)



(a) Multiplicative Attn for DenseNet-LSTM (b) Captioner Decoder and Filter Encoder used in BLIP

Figure 2: Attention for DenseNet-LSTM, Architecture for BLIP

with a recurrent LSTM-based text decoder. We used RoBERTa for context embeddings, which has been shown to outperform other common pretrained models such as BERT[15]. We implemented our models in PyTorch, adapting code from previous studies [8, 10]. Our modifications focused on two principal aspects: (1) the attention mechanism, and (2) modifications of the codebase to suit the requirements of our datasets. We changed the attention mechanism to use multiplicative attention instead of additive, as it has been shown to perform better on longer sequences [16].

The multiplicative attention can be represented as:

$$f_{\text{att}}\left(\mathbf{h}_i, \mathbf{s}_j\right) = \mathbf{h}_i^\top \mathbf{W} \mathbf{s}_j \tag{1}$$

where $\mathbf{W}$ is a matrix, $\mathbf{h_i}$ are the decoder hidden states and $\mathbf{s_j}$ are the encoder states. Our attention network is straightforward - separate linear layers reduce the encoded image from the Decoder to the same dimension, namely the Attention size. They are then *multiplied* and ReLU activated. A third linear layer reduces this result to a dimension of one, after which the softmax is used to generate the weights alpha. For the no-context model, the decoder receives all-ones vectors as context, which eventually get filtered out of the attention scores.

4.1.2 BLIP

BLIP [9] is a pretrained model from Salesforce Research which achieves state-of-the-art results on vision-language tasks, including image captioning. BLIP uses a multimodal encoder-decoder architecture, where BERT is the text encoder, and a vision transformer is the image encoder. Cross-attention asymmetrically combines two embedding sequences, where one serves as query Q and the other serves as key K and value V inputs.

$$f_{\text{att}}(q_i, k_j, V_j) = \sum_{j=1}^{N} \frac{\exp(q_i^\top k_j)}{\sum_{t=1}^{N} \exp(q_i^\top k_t)} V_j \tag{2}$$

where q_i is a specific query vector in matrix Q, k_j is a key vector in K, and V_j is a value vector in V where Q, K, V the query, key, and value embeddings respectively, and the RHS represents the attention weights computed based on the similarity between the query and key vectors. q_i is the i-th query vector in the decoder, k_j is the j-th key vector in the encoder, and N is the length of the encoder sequence [9].

BLIP uses an image-grounded text encoder (*filter*) and an image-grounded text decoder (*captioner*), where the encoder injects a cross-attention layer between the existing self-attention layer and the feed forward layer for each transformer block, and the decoder replaces the bi-directional self-attention layers with causal self-attention [9]. The decoder generates synthetic captions as additional training samples, and the encoder removes noisy captions that don't match their corresponding images, which helps BLIP bootstrap captions from noisy image-text pairs.

We fine-tuned the BLIP model with the Statista dataset, training both a no-context model and a model that incorporates context. For the no-context model, we use the image-description pairs as is, while for the context model, we concatenated the context embedding with the image embedding for the input to BLIP.

4.2 Baselines

Our baseline is the no-context model for each architecture, which takes in the image of the chart or graph and generates a summary. The with-context model takes in the image and the context, gets embeddings for each, and then generates a summary for the image of the chart or graph.

5 Experiments

5.1 Data

We combined and preprocessed datasets from Statista, Pew, Alt-Text HCI, and Concadia. We filtered the Concadia and HCI datasets using a script and visual inspection to extract charts and graphs, and included context for all datasets. Descriptive statistics about each dataset, such as vocabulary size and average length, are presented in Table 1. For HCI and Concadia, we used the *title* and available context as the actual context. For Pew, we used the underlying data tables extracted with OCR technology [2] as context - surrounding paragraphs were unavailable. For Statista, we used both the surrounding paragraphs and the underlying table (which was crawled in [3]) as context. Our original preprocessing procedure involved filtering, modifying context, and visual inspection, as indicated by the "Yes" descriptor under *Proc* in Table 1. See Appendix A for further analysis.

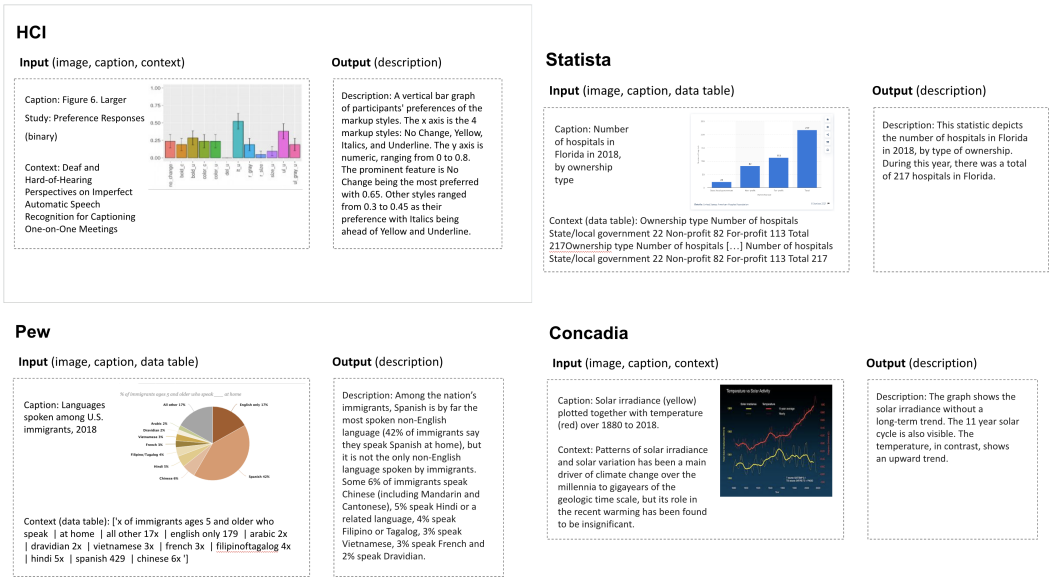


Figure 3: Examples from the datasets we worked with, with the described input/output.

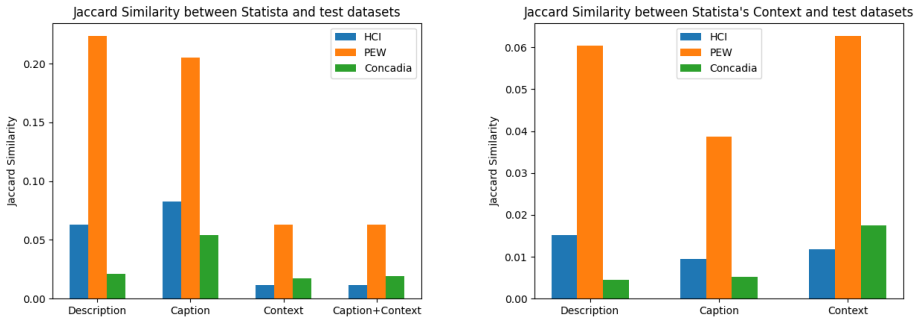
			Description			Caption			Context		
Dataset	**Images**	**Vocab**	Proc	avg size	wordlen	Proc	avg size	wordlen	Proc	avg size	wordlen
Statista	34811	232004	Yes	55.08	4.59	Yes	16.96	4.42	Yes	86.54	4.17
Pew	9285	46261	No	121.9	4.91	Yes	23.07	4.91	Yes	123.79	3.55
Concadia	398	6725	No	22.95	4.51	No	20.46	4.51	Yes	114.39	4.95
HCI	171	5938	No	99.31	4.04	No	33.33	4.45	No	45.21	4.63

Table 1: Datasets with Average Length and Word Size of Description, Caption and Context

5.1.1 Choice of Training and Test Data

For our second experiment, we chose Statista as our training dataset because it is the largest with $34,811$ data points and offers the best-defined context with underlying data tables and surrounding paragraphs. We used Pew, HCI, and Concadia as test datasets to investigate overfitting and model generalization. Jaccard similarity is a metric used to measure the similarity between two texts by

evaluating the number of common words between them. In Figure 4a, we do a 1-1 analysis of Statista's description, caption, context to others, and in Figure 4b zoom in to compare just the context to the test dataset. We hypothesize, using Jaccard similarities in Figure 4, that training on Statista makes it easier for our model to learn the relevant information and generate more accurate summaries for each of the test datasets.



(a) Statista's desc, caption, ctx compared (b) Statista's ctx compared to desc, caption, ctx

Figure 4: Jaccard similarity between Statista and test datasets

5.2 Evaluation method

Following previous work in charts and graphs, we evaluate with ROUGE, BLEU, and CIDEr scores [8, 3, 2]. All three metrics compare the summaries generated to human-authored references. The ROUGE score [17] is a metric used for automatic text summarization, which measures the number of overlapping words against a set of human referents, calculating precision and recall. The CIDEr score [18] measures how close a generated sentence is to a set of human-generated references by calculating the average cosine similarity between the candidate sentence and reference sentences for each n-gram.

We chose these metrics in order to compare to allow for comparison to existing literature. However, one limitation with these metrics is that we only have one human-authored reference for each image. To address this issue, in addition to recording ROUGE, BLEU, and CIDEr scores, we also conducted qualitative analysis where we measured the semantic distance from the ground truth using cosine similarity scores from SBert embeddings in Figure 5c.

5.3 Experimental details

We trained a total of 4 models - a no-context and context model each for (1) DenseNet-LSTM and (2) BLIP. For both architectures, due to **cost and memory constraints** which caused hurdles with our model training with our AWS VM, we trained on a subset of the Statista dataset ($17,405$ data points for DenseNet-LSTM and $11,613$ for BLIP), and tested on Pew, HCI, and Concadia.

For DenseNet-LSTM, we used a batch size 32, Adam optimization with cross-entropy loss, decoder learning rate of $4e-4$, and and a dropout coefficient of 0.5, and trained for 10 epochs. For BLIP, we reduced our vocabulary size from 30522 to 20522. For our hyperparameters of BLIP, we trained with a batch size of 4, used AdamW optimizer, a learning rate of $5e-5$, and trained it for 5 epochs.

5.4 Results

In Experiment 1, we see that DenseNet-LSTM outperforms ResNet on all metrics except for BLEU-1 on the no-context model, implying it is better suited for image captioning. However, both models perform poorly on the test split, indicating that they may be overfitting to the validation set and not generalizing well to new data. To test this hypothesis, we trained our model using Statista and tested on Pew, Concadia, and HCI to see how well our models generalize to unseen data.

In Experiment 2, we present two models - DenseNet-LSTM and BLIP. Comparing the results, we see that the BLIP model outperforms DenseNet on almost all metrics and datasets, albeit with a caveat -

Experiment 1: Evaluating models trained on all datasets

		BLEU-1		BLEU-4		CIDEr		ROUGE	
Split	Model	None	Ctxt	None	Ctxt	None	Ctxt	None	Ctxt
Val	ResNet	0.385	0.585	0.14877	0.360	0.48281	2.260	0.343	0.540
	DenseNet	0.4159	0.5799	0.1842	0.3558	0.7125	2.215	0.3774	0.5376
Test	ResNet	0.144	0.159	0.025	0.033	0.023	0.047	0.172	0.188
	DenseNet	0.134	0.1477	0.026	0.031	0.026	0.04	0.174	0.184

Experiment 2: Evaluating models trained on Statista on test datasets

Dataset		**BLEU-1**		**BLEU-4**		**CIDEr**		**ROUGE**	
	Model	None	Ctxt	None	Ctxt	None	Ctxt	None	Ctxt
Pew	BLIP	0.077	0.034	0.003	0.001	0	0	0.129	0.128
	D-LSTM	0.053	0.051	0.001	0.003	0.002	0.004	0.091	0.101
Concadia	BLIP	0.112	0.122	0.004	0.003	0	0	0.122	0.125
	D-LSTM	0.021	0.069	0.0	0.003	0.0001	0.007	0.020	0.076
HCI	BLIP	0.112	0.157	0.004	0.003	0	0	0.108	0.115
	D-LSTM	0.021	0.068	0.0	0.004	0.0001	0.031	0.016	0.094

it has a CIDEr score of 0 for all datasets. This could possibly be due to BLIP being pretrained on a newer, larger corpus of text, and was specifically designed for the purpose of image captioning, while DenseNet-LSTM comprises of two separate architectures - one specifically for image classification and one for sequence modelling. That being said, the performance of the finetuned BLIP is poor [1] - given our limited compute, we reduced our vocabulary size from 30522 to 20522, and trained with a low batch size of 4, which might have contributed to the substandard results.

Regarding the dataset, we can observe that the performance of our model varies across context types. For example, the Pew dataset performed worse with context, despite the Jaccard similarity in 4 being highest for Pew, while the Concadia and HCI datasets did better. We theorize that since the availability and quality of context varies across the datasets, this could affect model performance, and we plan to conduct further ablation studies.

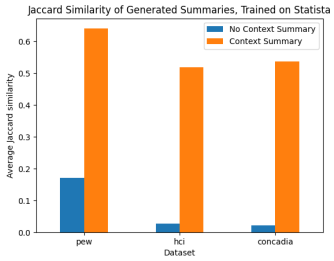
6 Analysis

We first examine how well context has been integrated with our models using Jaccard similarity. As an example, we pick hypotheses generated by DenseNet-LSTM, and observe in Figure 5a that our hypotheses from context models are significantly closer to our context than our hypotheses from no-context models, which is also observed in 5c. We have the best scores for Pew, as examined in Figure 5a and Table 2, followed by Concadia and then HCI. While our ROUGE score trend confirms this, our CIDEr score reflects an inverse trend of this order. This could be explained by how these metrics are calculated - while ROUGE and Jaccard both measure the similarity using overlapping n-grams, CIDEr relies on multiple references labels and takes into account novelty of the generated summaries [18], which we lack. We speculate that context models are focusing on the underlying salient information and not exploring alternative ways of generating it. Further analysis of trends in our hypotheses has been done in Appendix A.

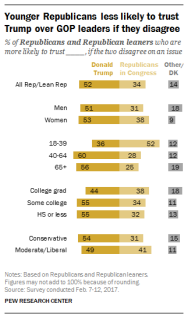
DenseNet-LSTM performs poorly in Experiment 2. Since our summaries are long, one possibility is that our LSTM model is unable to capture long-term dependencies in text. There could also be noise in the Statista dataset, which may have affected the model's ability to learn relevant information - DenseNet-LSTM's dense connectivity may be overfitting. One possible mitigation is to have a

[1] Our CIDEr score of 0 may possibly due to only having one reference label, non-generalizability of the test dataset, or human error.

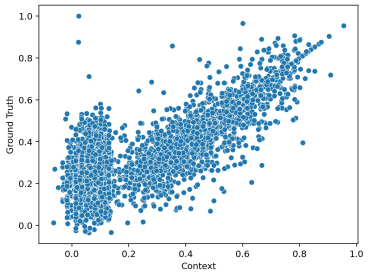
human-subject experiment where we filter out datapoints that do not have a relevant summary and context. Another choice is to use a more complex architecture - e.g. with BLIP in Experiment 2.



(a) DenseNet-LSTM: Jaccard Similarity of our contexts with our hypotheses from both models



(b) Ref image for Pew example in Table 4



(c) Correlation of the hypotheses generated by the DenseNet-LSTM context model (in Experiment 1). On the x-axis: semantic distance to context, y-axis: semantic distance to ground truth. We observe a weakly positive correlation.

Figure 5: A deeper dive into DenseNet-LSTM's results

Original Label	No Context	Context
On balance, Republican and Republican leaners are more likely to side with Donald Trump over Republican leaders if there is a disagreement between them on an issue. About half (52%) say they would be more likely to trust Donald Trump, while only about a third (34%) say they would be more likely to trust Republican leaders in Congress.	This statistic shows the number of VAT and PAYE based enterprises in the United Kingdom (UK) in the second quarter of 2020, by age. As of the second quarter of 2020, there were approximately 1.15 million households in this sector that year.	This statistic shows the results of a survey of respondents from the United States on whether they think that they would be willing to choose to vote.

Table 2: DenseNet-LSTM: Example from Pew with original summary, label without context, label with context. We see how the no-context summary has been overfitted, which reflects in our results from Experiment 2, while the context summary is more general and reflects the overarching context in which the original summary is written.

7 Conclusion

7.1 Main findings

In our first experiment, we used ResNet-LSTM and DenseNet-LSTM to generate summaries. For our second experiment, we used DenseNet-LSTM and BLIP, and trained two models for each - with and without context. We used an encoder-decoder model with attention to generate summaries. We hypothesized Statista would generalize well based on its context, so we chose it as the training dataset and the others as test datasets. We used BLEU-1, BLEU-4, CIDEr and ROUGE metrics to evaluate the model trained on Statista, and conducted further analysis using Jaccard similarity and cosine similarities of embeddings. We found that context improves the quality of generated summary in every model. Although training on Statista resulted in slightly lower scores than on the concatenated datasets without context, training with context generalizes well to the other datasets.

7.2 Limitations and Future Work

Due to the limitations of our dataset, each chart/graph has only a single human-written reference for it, which impacts our CIDEr scores and the generation of natural language summaries. Secondly, there is currently no metric to standardize all four of the chart/graph datasets. In the future, we plan to conduct human subject experiments to author more references and filter the datapoints that are relevant for our project. We also plan to conduct more ablation studies with different models, datasets and hyperparameters, to mitigate issues like overfitting.

References

[1] Sanjana Shivani Chintalapati, Jonathan Bragg, and Lucy Lu Wang. A dataset of alt texts from HCI publications. In *The 24th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, oct 2022.

[2] Jason Obeid and Enamul Hoque. Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model. In *Proceedings of the 13th International Conference on Natural Language Generation*, 2020.

[3] Shankar Kantharaj, Rixie Tiffany Ko Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. Chart-to-text: A large-scale benchmark for chart summarization. In *Association for Computational Linguistics (ACL)*, 2022.

[4] Dae Hyun Kim, Enamul Hoque, and Maneesh Agrawala. Answering questions about charts and generating visual explanations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery.

[5] Charles Chen, Ruiyi Zhang, Eunyee Koh, Sungchul Kim, Scott Cohen, Tong Yu, Ryan Rossi, and Razvan Bunescu. Figure captioning with reasoning and sequence-level training, 2019.

[6] Abigale Stangl, Meredith Ringel Morris, and Danna Gurari. Person, shoes, tree. is the person naked? what people with vision impairments want in image descriptions. In *Conferences on Human-Computer Interaction*, 2020.

[7] Abigale Stangl, Nitin Verma, Kenneth Fleischmann, Meredith Ringel Morris, and Danna Gurari. Going beyond one-size-fits-all image descriptions to satisfy the information wants of people who are blind or have low vision. In *23rd International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '21)*, 2021.

[8] Elisa Kreiss, Fei Fang, Noah Goodman, and Christopher Potts. Concadia: Towards image-based text generation with a purpose. In *Empirical Methods for Natural Language Processing*, 2022.

[9] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022.

[10] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention, 2015.

[11] Alan Lundgard and Arvind Satyanarayan. Accessible visualization via natural language descriptions: A four-level model of semantic content. In *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[13] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

[14] Xiaofeng Mao, Yuefeng Chen, Xiaojun Jia, Rong Zhang, Hui Xue, and Zhao Li. Context-aware robust fine-tuning, 2022.

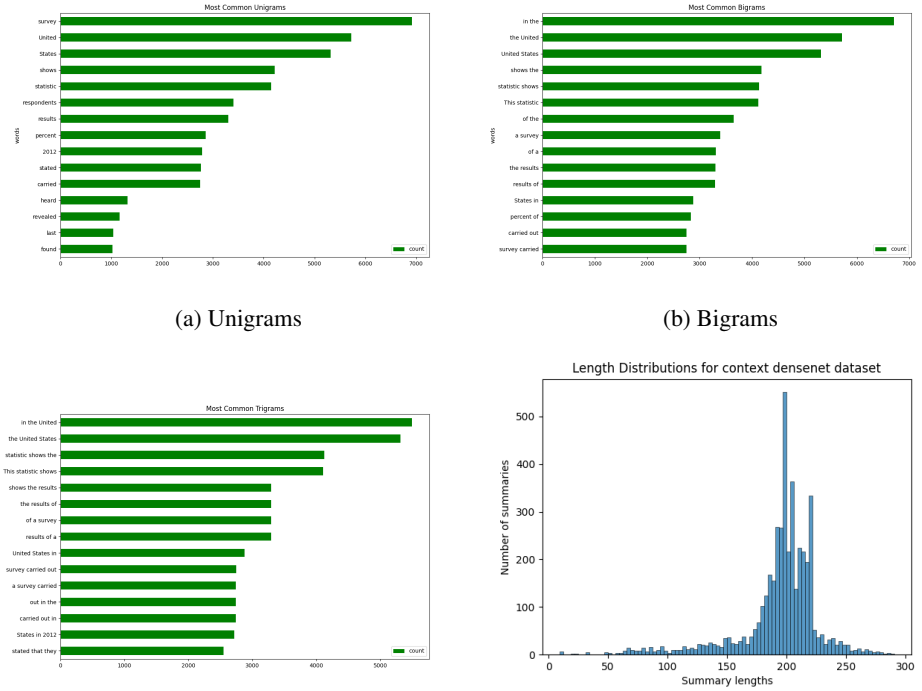
[15] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[16] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation, 2015.

[17] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2004.

[18] Ramakrishna Vedantam, Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4566–4575*, 2015.

A Appendix



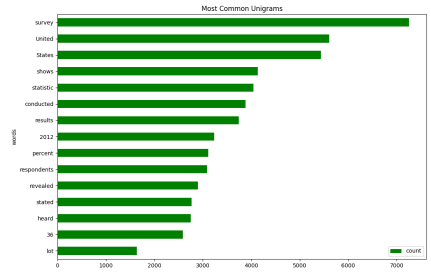
(a) Unigrams

(b) Bigrams

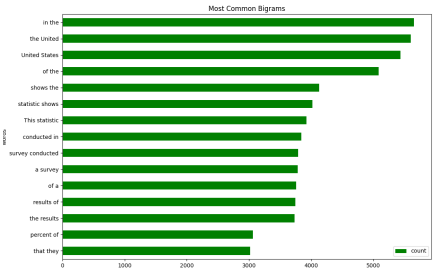
(c) Trigrams

(d) Length

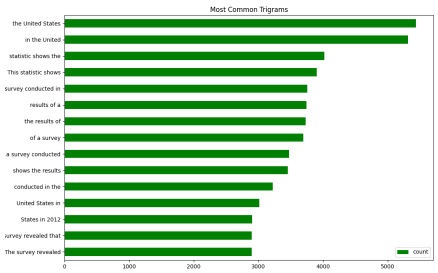
Figure 6: Analysis of Predictions by DenseNet-LSTM with Context



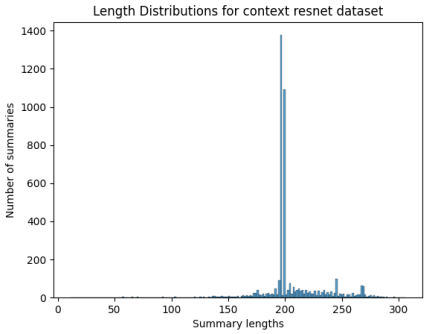
(a) Unigrams



(b) Bigrams

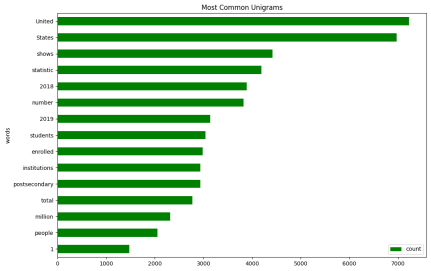


(c) Trigrams

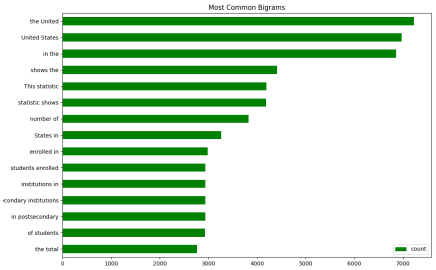


(d) Length

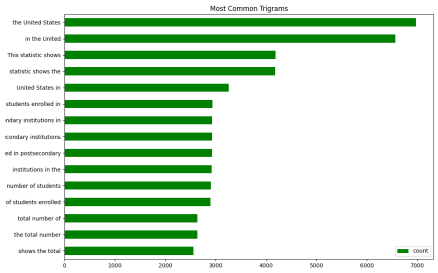
Figure 7: Analysis of Predictions by ResNet-LSTM with Context



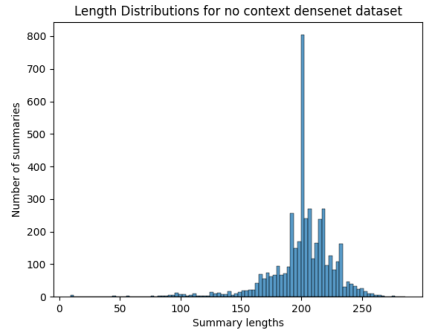
(a) Unigrams



(b) Bigrams

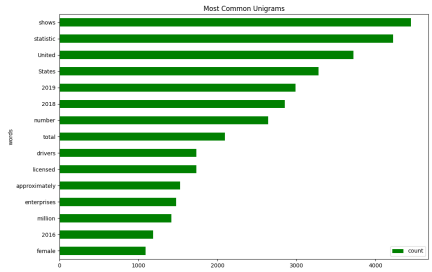


(c) Trigrams

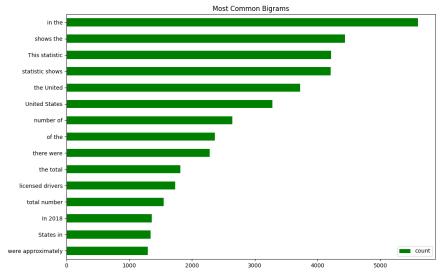


(d) Length

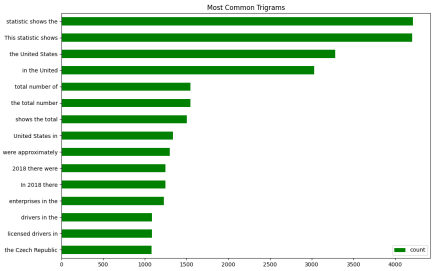
Figure 8: Analysis of Predictions by DenseNet-LSTM with No Context



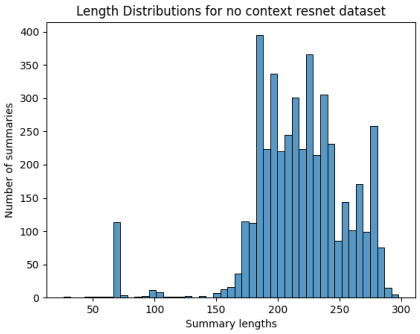
(a) Unigrams



(b) Bigrams

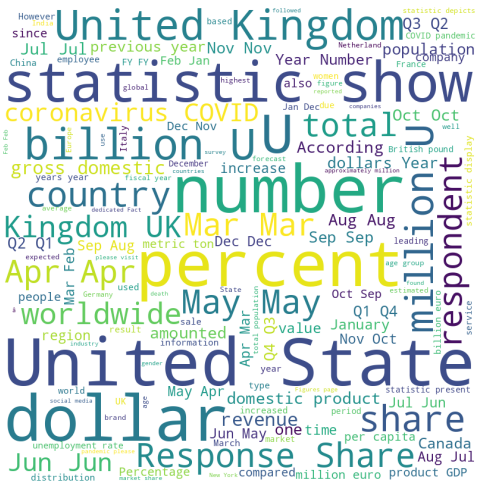


(c) Trigrams



(d) Length

Figure 9: Analysis of Predictions by ResNet-LSTM with No Context



(a) Statista (b) Pew

(c) HCI (d) Concadia

Figure 10: Wordclouds for our datasets