From Data to Insight: Exploring Program-of-Thoughts Prompting for Chart Summarization

Anonymous ACL submission

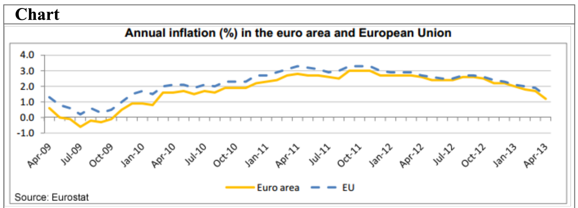
Abstract

Charts play a critical role in conveying numerical data insights through structured visual representations. However, semantic visual understanding and numerical reasoning requirements hinder the accurate description of charts, interpreting a challenging task in chart summarization. Despite recent advancements in visual language models (VLMs), approaches lack robust mechanisms for verifying statistical fact correctness and are computationally heavy. To address this gap, this paper explores the potential of using zero-shot learning to motivate the lightweight VLMs to perform computational reasoning via Python programs as intermediate outputs to derive summary statistics valid for chart understanding. Specifically, we introduce a novel chart-to-dictionary auxiliary task, offering a more flexible representation compared to traditional chart-to-table methods, making it particularly well-suited for integration with the Program-of-Thought (PoT) strategy. Experimental results demonstrate that our method performs on par with existing chart summarization methods across machine translation and text generation metrics. We release the code at the GitHub link.

1 Introduction

With the rising demand for visualizing quantitative data, the growing adoption of digital media has played a role in the rapid growth of data visualization, which has led to the task of automatic chart understanding, information extraction, and summarization, critical areas of research (Huang et al., 2024; Zhang et al., 2024; Choi et al., 2025). Recent advancements in Visual Language Models (VLMs) have shown promise in this area (Masry et al., 2023; Han et al., 2023; Ko et al., 2024; Masry et al., 2024; Meng et al., 2024; Zhang et al., 2024; Liu et al., 2024); however, existing methods still struggle with achieving high-quality summaries, especially for L2/L3 content - which is identified as statistics and relations (e.g., min, max) / perceptual and cognitive phenomena (e.g., trends) (Lundgard and Satyanarayan, 2022; Kantharaj et al., 2022; Tang et al., 2023), as shown in Figure 1. The challenge is around the highly inconsistent matching between the generated summary and the chart's actual data content yields factual inconsistencies and hallucinations. This is either due to failing to parse the text in the chart or to demarcate the numerical value of the visualized data. Additionally, with semantic parsing of the chart elements, VLMs struggle at performing complex reasoning about chart patterns and incorporating statistical reasoning with chart elements (Liu et al., 2024). Despite general challenges, although current VLM-based chart understanding methods have shown a certain level of performance, they still face two main challenges: (1) Existing implementations are fine-

Chart



Summary

Eurostat, the European statistical agency, announced today that the European Union's annual inflation rate in *April was 1.4%, down from a rate of 2.7% in April 2012*. Nonetheless, across eight EU nations surveyed by the Pew Research Center in March 2013...

Dictionary

{"Euro area": [{"date": "Apr-09", "value": 1}, {"date": "Jul-09", "value": ~~-2~~}, ..., "EU": [{"date": "Apr-09", "value": 1}, {"date": "Jul-09", "value": ~~-1~~}, ... , {"date": "Jan-13", "value": ~~14~~}, {"date": "Apr-13", "value": ~~15~~}]}

Statistics

{"Euro area": {"minimum_value": ~~-2~~, "maximum_value": ~~16~~, "total_sum": ~~132~~, ... "EU": {"minimum_value": ~~-2~~, "maximum_value": ~~16~~, "total_sum": ~~132~~, "number_of_entries": 17, "average_value": ~~7.76~~...}}

Generated Summary

Both the Euro Area and EU experienced an increase from April 2009 through January 2013, ~~*reaching their highest points at around 16% by mid-January 2-013*~~ for both regions...there was a gradual decline until May or June that year before stabilizing between approximately 1.5%-2%...

Figure 1: Example of a chart in Pew with its data representations in Python dictionary and statistics. *Italic* is the L2/L3 content in the chart summarization. ~~Strikeout~~ indicates hallucination errors and error-inducing tokens.

tuned or pre-trained specifically on chart-related instruction data. While this alignment between the vision encoder and language decoder enhances generalization performance, such training processes introduce significant computational overhead, making them resource-intensive and challenging under computational constraints; (2) These tasks continue to remain a challenge in understanding the structural interplay between the different elements of a chart. Effective visual language understanding in particular requires two key processes: first, comprehensive semantic layout understanding of the chart, and second, robust statistical reasoning to accurately capture and analyze the underlying data (Liu et al., 2023b). In light of these challenges, we investigate zero-shot and training-free approaches for VLMs in chart summarization, exploring whether supplementary textual data in multi-modal chart summarization enhances or hinders overall performance to what extent. Program-of-Thoughts (PoT) (Chen et al., 2023) is a zero-shot prompting method, which was originally proposed to disentangle computation from reasoning to augment a model's numerical capability. PoT has shown effectiveness in enhancing the ability of language models compared to the general multimodal-purpose prompting Multimodal Chain of Thought (MCoT) (Wang et al., 2025) in complex numerical reasoning tasks. Given this statistical reasoning capability, this work investigates the effectiveness of the PoT guiding VLMs to perform numerical computations and logical reasoning via LLM Python programs as intermediate steps in the chart summarization process. VLMs will be used to generate summaries in zero-shot settings with the PoT approach for the chart summarization task. Specifically, we aim to answer the following research questions centering on chart summarization: **RQ1.** *How can visualized numerical data, such as charts, be represented using Program-of-Thoughts prompting?* **RQ2.** *Does offloading statistical computations from a VLM improve its performance for concluding the L2/L3 content?* and **RQ3.** *How do Program-of-Thoughts prompting improvements affect different chart types, in terms of area, line, bar, pie, and scatter charts?*

This work integrates the PoT methodology on data representation for chart summarization with VLMs, serving as the chart data representation in chart understanding to aid in the summarization task. We also demonstrate that PoT offers a competitive performance relative to existing prompting methodologies in the context of lightweight VLMs.

2 Literature Review

2.1 Chart Understanding

Template-Based Early approaches to automatic chart understanding, particularly the sub-task of chart summarization, often relied on planning-based architecture and template-based generation methods (Mittal et al., 1998; Fasciano and Lapalme, 2000; Green et al., 2004; Reiter, 2007; Ferres et al., 2007, 2013). Recent template-based research has focused on utilizing statistics (e.g., min, max, trends) from chart numerical data for presenting the facts (Demir et al., 2012; Cui et al., 2019; Srinivasan et al., 2019; Wang et al., 2020), forming the statistics analysis into textual summarization output. Some research utilized the off-the-shelf OCR (Optical Character Recognition) tools or detectors to represent chart data into textual tables and other representations, relying on pipeline methods (Singh et al., 2019; Sidorov et al., 2020; Methani et al., 2020; Hu et al., 2021; Fu et al., 2022; Kantharaj et al., 2022; Liu et al., 2023a). More recently, ResNet (He et al., 2016) encoder and LSTM decoder were used to process the chart and create the caption (Chen et al., 2020a). However, compared to data-driven models, template-based approaches struggle with complex visual patterns and numerical reasoning, with high costs in producing generics and matching variations in vocabulary choices.

Pretrained With the progression of deep learning techniques, which subsequently improved general computer vision using neural networks and Transformer (Vaswani et al., 2017), recent work began to adopt encoder-decoder architectures to improve chart understanding (Wang et al., 2025), including Transformer (Singh and Shekhar, 2020; Obeid and Hoque, 2020; Kantharaj et al., 2022; Lee et al., 2023), LSTM (Spreafico and Carenini, 2020), CNN+LSTM (Hsu et al., 2021), and VLMs (Liu et al., 2023b), which are pre-trained on both visual and text data, often with specialized text and image encoders, and have shown significant promise in tasks requiring joint understanding of multiple modalities. However, challenges remain in grounding the factual and logical coherence in generated summaries, particularly when dealing with complex charts requiring numerical reasoning.

Fine-Tuned Aside from pre-training the model, fine-tuning the pre-training model (Tang et al., 2023) and instruction fine-tuning (Ouyang et al., 2022) have also become widely adopted as an alternative to improve the performance of LLMs and

VLMs (Masry et al., 2023; Han et al., 2023; Ko et al., 2024; Masry et al., 2024; Meng et al., 2024; Zhang et al., 2024; Liu et al., 2024). Instruction tuning is used to generalize the language capability of the model, reducing repetitions and hallucinations generated in summarization than pre-training approaches (Meng et al., 2024). However, these methods typically rely on the data tables of charts, failing to capture the nuance of the visual artifacts present in charts. Furthermore, their heavy parameter sizes present notable challenges for deployment in computationally constrained environments.

2.2 Chart Representations

Representing the chart in structured data, the chart-to-table (Meng et al., 2024) task represents it in the tabular format, but often comes at the cost of losing finer details in the chart. Performing similarly to data tables, scene graphs are easily formatted for web-based charts (Tang et al., 2023). Code format is considered, and existing methodologies define two typical chart-to-code approaches: (1) Chart Derendering (Liu et al., 2023b; Lee et al., 2023); and (2) Program of Thoughts (Chen et al., 2023; Zhang et al., 2024). However, codes mainly aim to run for the chart recreation or question answering tasks on narrowly defined questions, rather than representing the whole chart. This paper proposes an auxiliary task of chart-to-table, which is chart-to-dictionary in Python code format, which uses VLM's chart understanding capability to represent the chart as a Python dictionary.

2.3 Prompting

Inspired by the success of Chain-of-Thought (CoT) prompting (Wei et al., 2022) for improving reasoning capabilities, researchers are extending similar mechanisms to VLMs for chart understanding, seeking to mirror the human cognitive process of visual analysis. This is achieved through MCoT (Wang et al., 2025; Liu et al., 2024) reasoning, which extends the rationale from texts to visual modalities (Choi et al., 2025). To contrast with MCoT, PoT (Chen et al., 2023; Zhang et al., 2024) prompting intermediate reasoning steps are articulated as executable programs, while executing the program to generate reasoning and statistical computation about the chart data. The success of PoT in chart question answering (QA) has motivated our exploration of chart summarization, which focuses on generating more structurally complex and extensive sentences, rather than just concise answers.

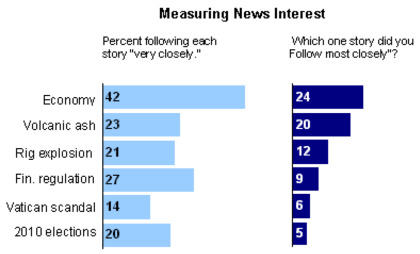
In this work, our pipeline method builds upon these advancements by focusing on PoT prompting in zero-shot chart summarization. By extending the PoT concept to the visual domain of charts, it could decrease hallucinations that language models typically have when outputting calculations, as it provides more explicit and verifiable numeric reasoning processes for VLMs (Zhang et al., 2024), potentially leading to more accurate and factually grounded summaries by delegating complex calculations to a code interpreter. This work differentiates itself from existing works by specifically investigating the benefits and limitations of generating executable code as intermediate reasoning steps for chart summarization with lightweight VLMs.

3 Method

This paper proposes using PoT to augment a VLM's capability for statistical reasoning on chart data generation, as one of the key elements for visual language reasoning (Liu et al., 2023b). Exploring the model's capacity for zero-shot setting for reasoning, this paper adapts a similar methodology described in PoT, as the work (Chen et al., 2023) stated that the program's line-by-line structure acts as a proxy for the numerical reasoning steps of the model. Similar to the previous work, the usage of '#' tokens in the generated tokens was restricted to avoid the pitfalls of only generating the reasoning chain as comments instead of executable code. Our prompts are illustrated in Appendix C.

3.1 Chart Representation as a VLM-Generated Python Dictionary

In order for the chart to interface with the code, the chart needs to be represented in a manner that can interact with the Python interpreter. As shown in Figure 2, Python dictionaries can represent the code in a more free-form structure, allowing for grounding the values compared to the data table. However, lightweight VLMs can struggle to create executable Python code, which consists of wrong syntax, incomplete messages, and even meaningless code-agnostic terminologies when facing the complex code generation request, adding noise. Given that, aside from reflecting understanding from charts, the code needs to be valid and executable. In Appendix E, we list more details of the failure case analysis. To handle failure cases in dictionary generation, we mainly used InternVL-2.5-4B (Chen et al., 2024) to do this task in a zero-shot setting,



{"Economy": {"Very closely": 42, "Most closely": 24}, "Volcanic ash": {"Very closely": 23, "Most closely": 20}, "Rig explosion": {"Very closely": 21, "Most closely": 12}, "Fin. regulation": {"Very closely": 27, "Most closely": 9}, "Vatican scandal": {"Very closely": 14, "Most closely": 6}, "2010 elections": {"Very closely": 20, "Most closely": 5}}

Figure 2: Representing chart (left) as a Python dictionary (right). Python dictionary representation is more flexible compared to a markdown table and usable by the LLM-generated program.

and if the generated Python dictionary is not executable, it is converted with ChatGPT (GPT-4o-mini) (OpenAI, 2024) instead.

3.2 Statistical Analysis with PoT prompting

Since the chart is represented as a Python dictionary, it can be more free-form in containing data and being passed to a Python program. Code is passed to an LLM to generate a program to do statistical analysis as an intermediate result to provide more context for chart summarization. Compared to QA as a task, statistical analysis with PoT demonstrates numerical reasoning since it demonstrates how the models understand which data points or statistics are necessary to create summary statistics. This paper uses Qwen-2.5-Coder-14B (Hui et al., 2024) for the complex statistics code generation conversion, with a case study and pipeline presented in Figure 3. The LLM is instructed to generate a Python program using the Python dictionary in the prompt to generate summary statistics relevant to the chart dictionary. This adapts PoT for the chart summarization task as the generated program provides more context to be used for text generation while providing accurate calculations. Code generated by the LLM is constrained to use only the functions from Python's built-in library. To validate and execute the generated Python program by the PoT strategy, we used the built-in `exec` function in Python for automatic code validation.

3.3 Program Execution and Statistics Retrieval

The generated Python program for statistics calculation is executed using a Python interpreter. This step ensures the accuracy of the statistical results, mitigating potential errors that LLMs might make when generating tokens through direct calculations. The program returns a Python statistics dictionary that contains key-value pairs of the summary statistics and the calculated values. At the end, the statistical results in our pipeline are input with the chart into a VLM to assist the chart summarization task.

Type	**Pew**			**VisText**		
	Simp.	Comp.	All	Simp.	Comp.	All
Area	13	7	20	157	81	238
Bar	840	128	968	304	127	431
Line	312	37	349	135	78	213
Pie	41	0	41	0	0	0
Scatter	0	15	15	0	0	0
Total	1,206	187	1,393	596	286	882

Table 1: Distribution of chart types by **Simp**le and **Comp**lex complexities of the Pew and VisText datasets.

4 Experiment

We present our experimental setup in Appendix A. The overview of our datasets, evaluation metrics, baseline methods, and benchmark and backbone models is provided in the following subsections.

4.1 Baselines and Evaluation

Evaluation. We evaluated PoT prompting for chart summarization on both the test sets of the Pew (Kantharaj et al., 2022) and VisText (Tang et al., 2023), following the previous evaluation works (Masry et al., 2023; Meng et al., 2024) for evaluating the PoT on varying degrees of complex charts to show its generalizability. The VisText is built upon Statista (Kantharaj et al., 2022) with richly labelled L2/L3 captions. Chart type distributions of datasets are summarized in Table 1, which across a variety of simple and complex charts. More details on the dataset statistics and topic distribution information are presented in the Appendix B.1. To evaluate the effectiveness of the methods, we employ BLEU (Post, 2018) and CIDEr (Vedantam et al., 2015) as the evaluation metric following previous works (Kantharaj et al., 2022; Liu et al., 2023b; Masry et al., 2023; Meng et al., 2024). Addition-

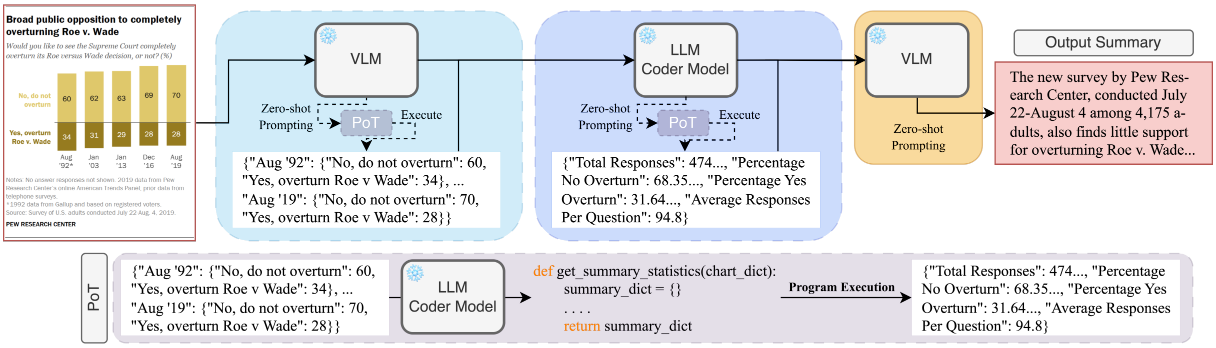


Figure 3: Process of implementing the Program of Thought (PoT) given a chart. It can be seen as a process of enhancing statistical reasoning to extract summary statistics, typically total counts, minimum, and maximum values from the chart, along with labels that contain the numerical values.

ally, we use F1 scores of ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2020) for evaluation. ROUGE is a prevailing benchmark in text summarization research, whereas BERTScore offers a complementary perspective by quantifying semantic similarity between system outputs and reference texts. We provide more details of evaluation metrics in the Appendix B.3.

Baselines. We compare two other types of prompting strategies as baselines: (1) Directly prompting (Direct) the model to summarize the chart, given that this approach is also what is done by fine-tuned end-to-end models (Huang et al., 2024; Liu et al., 2024); (2) Multimodal CoT (MCoT), which adheres to the framework in (Wang et al., 2025), prompting to return an outline of all key information and trends derived from the chart.

4.2 Benchmarks and Backbones

Chart-To-Text Models. To assess the effectiveness of our PoT prompting approach against existing models and methods in the chart-to-text domain, we choose: (1) Pretrained Chart-To-Text: OCR-Field-Infuse (Chen et al., 2020b; Kantharaj et al., 2022), Monkey (Li et al., 2024); (2) Prefix-tuning Chart-To-Text: image-scene-graph-PT (Tang et al., 2023), image-data-table-PT (Tang et al., 2023); (3) Commonly used VLMs: Blip2-flant5xl (Li et al., 2023), Qwen-VL (Bai et al., 2023).

VLM Models. To understand the effects of PoT on the different VLM backbones, we compare the performance of Deepseek-VL2-tiny (Wu et al., 2024), InternVL-2.5-4B (Chen et al., 2024), LLaVa-v1.6-mistral-7B-hf (Liu et al., 2023c), and Qwen2.5-VL-3B-Instruct (Qwen Team, 2025) on the representative datasets of Pew and VisText. All experiments were done with the zero-shot setting models.

Method	Pew		VisText	
	BLEU	CIDEr	BLEU	CIDEr
OCR-Field-Infuse	0.2	0.3	0.3	-
Monkey	0.4	1.7	-	-
Qwen-VL-9.6B	0.5	2.6	-	-
Blip2-flant5xl-4B	0.2	0.8	-	-
image-scene-graph-PT	-	-	0.3	-
image-data-table-PT	-	-	0.3	-
Qwen2.5-VL-3B+PoT	**3.1**	0.1	**1.7**	0.1

Table 2: We compare our PoT-adopted zero-shot VLM (Qwen2.5-VL-3B+PoT) with different chart summarization methods on Pew and VisText test datasets. We referenced the results from Chart-To-Text (Kantharaj et al., 2022), VisText (Tang et al., 2023), and ChartAssistant (Meng et al., 2024).

5 Evaluations of Our PoT Approach Against Existing Benchmarks

Table 2 shows the BLEU and CIDEr scores for each model on the Pew and VisText datasets. We referenced evaluation results from Chart-To-Text (Kantharaj et al., 2022), VisText (Tang et al., 2023), and ChartAssistant (Meng et al., 2024). As shown in the table, we observed that our PoT prompting approach overperforms baseline Chart-To-Text methods in the BLEU evaluation scores, but underperforms in the CIDEr evaluation scores. This may be due to CIDEr places more emphasis on important and rare words, as it calculates TF-IDF weighted n-gram similarity. While BLEU also focuses only on surface-level word matching and ignores semantic consistency, we subsequently evaluate our PoT prompting approach on BERTScore and ROUGE-L evaluation metrics, as they can better consider and evaluate semantic information. More details of experimental results and additional ablation studies are in the Appendix B.4.

VLM & Prompting	Pew												VisText							
	Area		Bar		Line		Pie		Scatter		All		Area		Bar		Line		All	
	BLEU	CIDEr	BLEU	CIDEr	BLEU	CIDEr	BLEU	CIDEr	BLEU	CIDEr	BLEU	CIDEr	BLEU	CIDEr	BLEU	CIDEr	BLEU	CIDEr	BLEU	CIDEr
deepseek-vl2-tiny																				
ZeroShot-Direct	1.9682	0.0427	2.6653	0.0608	1.7169	0.0471	4.5805	0.1391	0.7646	0.0412	2.4676	0.0591	1.8347	0.0920	1.5262	0.0731	2.0429	0.0851	1.7346	0.0824
ZeroShot-MCoT	1.6352	0.0526	1.8918	0.0403	1.2924	0.0360	3.1608	0.0671	1.0925	0.0657	1.7658	0.0399	0.9308	0.0410	0.7613	0.0353	1.1508	0.0388	0.9001	0.0380
ZeroShot-PoT	0.1254	0.0018	0.2767	0.0127	0.2736	0.0173	0.2496	0.0190	0.2219	0.0005	0.2746	0.0135	0.8102	0.0710	0.3489	0.0523	0.5821	0.0685	0.5603	0.0615
internVL-2.5																				
ZeroShot-Direct	3.6507	0.0426	3.5832	0.0318	2.7521	0.0296	4.6431	0.1025	2.6224	0.0001	3.4041	0.0328	1.1306	0.0125	0.9387	0.0088	1.3401	0.0212	1.0808	0.0130
ZeroShot-MCoT	2.3817	0.0257	2.0626	0.0106	1.4369	0.0061	1.9856	0.0053	1.5318	0.0003	1.9113	0.0094	0.8414	0.0022	0.8978	0.0005	1.0359	0.0030	0.9175	0.0015
ZeroShot-PoT	2.8535	**0.0713**	1.9995	**0.0664**	1.9136	**0.0404**	2.0840	0.0907	1.3768	**0.0819**	1.9896	**0.0603**	1.1281	**0.0246**	0.9299	**0.0172**	**1.6892**	**0.0274**	**1.1736**	**0.0212**
llava-NeXT																				
ZeroShot-Direct	4.8807	0.1561	5.7756	0.1069	4.6735	0.1133	7.8216	0.2135	4.3993	0.0074	5.5350	0.1107	2.6597	0.0272	2.5564	0.0334	3.4469	0.0612	2.7918	0.0384
ZeroShot-MCoT	6.1606	0.0329	5.9175	0.0928	4.6181	0.0644	5.7460	0.1498	3.9118	0.0808	5.6347	0.0869	2.5957	0.0478	2.2776	0.0243	3.5833	0.0499	2.6622	0.0365
ZeroShot-PoT	3.1421	0.1069	4.1897	0.1027	3.5534	0.0925	2.7975	0.0895	3.3424	**0.1210**	3.9888	0.0996	2.3603	0.0321	2.2635	**0.0457**	2.9584	0.0580	2.4604	**0.0448**
qwen2.5-VL-3B																				
ZeroShot-Direct	1.9350	0.0523	3.6251	0.1002	2.5562	0.0643	5.9420	0.1384	2.0714	0.0272	3.3929	0.0905	2.6399	0.1481	2.1772	0.0979	3.1147	0.1519	2.4984	0.1254
ZeroShot-MCoT	1.4980	0.0735	2.6168	0.0814	1.8583	0.0602	3.7722	0.2156	1.5976	0.0431	2.4388	0.0794	1.5847	0.0837	1.3648	0.0791	1.9742	0.0707	1.5783	0.0782
ZeroShot-PoT	**3.3383**	0.0409	3.3091	0.0734	2.3678	0.0597	3.8250	0.1662	1.0761	0.0203	3.0906	0.0712	1.6593	0.0780	1.4806	0.0801	2.0928	0.0890	1.6639	0.0826

Table 3: Evaluation results of VLMs on different prompting methods on Pew and VisText datasets evaluated on BLEU and CIDEr scores.

6 Evaluations of Our PoT Approach Against Existing Baselines

We evaluate baseline prompting strategies and our PoT prompting strategy and report results of our experiment in Table 3. Across the evaluated models, the impact of the PoT prompting strategy varied significantly with models and chart types. We observed instances where the PoT led to substantial improvements in performance, while in other cases, its impact was less pronounced or even negative compared to the Direct and MCoT approaches.

PoT Effectiveness Against Chart Types We notice that the results from different charts are varied, and we suppose this may be due to the uniqueness of each chart structure, texts included in the chart, chart data size, and data complexity. For example, in the case of the Qwen2.5-VL model, the evaluation score increases from 1.935 to 3.338, demonstrating the effectiveness of the PoT strategy in enhancing information collection from area charts, which are with limited data information.

PoT Effectiveness Against VLMs Regarding influences by VLMs, for the Deepseek-vl2-tiny model, the application of the PoT resulted in considerably lower scores across all reported metrics compared to both the ZeroShot-Direct and ZeroShot-MCoT methods. This suggests that for this particular model architecture, the PoT strategy in its current implementation might not be beneficial or could even hinder performance on the evaluated tasks. This reveals that the PoT strategy may introduce additional noise or mislead the emphasized information, and may interfere with the model's original processing and understanding of the chart. In contrast, the InternVL-2.5 model demonstrated a more nuanced response to the PoT prompting strategy. While the Direct method often yielded the highest scores, the PoT strategy achieved comparable or even slightly better results on certain metrics compared to the MCoT strategy in some cases. For example, the PoT strategy achieved a BLEU score of 2.854, which is lower than the Direct method (3.651) but higher than the MCoT strategy (2.382) of the area charts in the Pew dataset, even on considering all chart types, these trends hold. This indicates that for InternVL-2.5, the PoT strategy can be a viable alternative to the MCoT strategy in certain scenarios, although the Direct prompting method appears to be generally more effective based on these results. Similarly, LLaVa-NeXT also had a mixed response given the two datasets, where no conclusive trends can be observed between the different prompting methods. One interesting observation from this comparison is that while the BLEU values of the PoT strategy are lower than the other methods, on average, it outperforms the other prompting techniques on CIDEr, indicating some of its effectiveness in these cases. We suggest that this may arise from the inherent design differences in the VLMs with respect to chart understanding. Specifically, Deepseek-vl2 is equipped with a dedicated vision encoder and a vision-language adapter, originally designed to optimize performance on visual tasks such as chart interpretation. In contrast, InternVL-2.5 is built upon a Vision Transformer architecture integrated with a large language model, placing more confidence on the fusion of textual information. As a result, when we enlarge the textual data using the PoT strategy, the performance outcomes of Deepseek-vl2 and InternVL-2.5 can diverge, potentially yielding opposite trends. This observation suggests that the PoT strategy does not universally benefit all VLMs in chart summarization, but is particularly advantageous for those that emphasize textual information.

PoT Compared with MCoT On the other hand, Qwen-2.5VL-3B showed that the PoT strategy consistently outperformed the MCoT strategy while underperforming relative to the Direct prompting. This suggests that for the Qwen2.5-VL-3B model, the PoT strategy appears to be a more effective CoT prompting strategy compared to the standard MCoT approach across the evaluated tasks. This may be due to the PoT strategy introducing more new textual content into the chart summarization process compared to the MCoT approach. While the PoT generates additional statistical information, MCoT primarily offers a high-level data outline and trends. The PoT may introduce additional noise and errors, particularly due to inaccuracies in chart data interpretation by the InternVL-2.5 model.

While the PoT strategy demonstrated potential for improving performance, particularly for the InternVL-2.5 and Qwen2.5-VL-3B models in certain scenarios, it did not consistently outperform the Direct prompting baseline across all models and metrics. Further investigations are conducted to identify the factors contributing to the varying effectiveness of the PoT strategy and to estimate the extent to which different information influences the performance of this pipeline.

7 Evaluations of Our PoT Approach Against VLM Backbones

To investigate the influence of different types of information within the PoT strategy pipeline, we conducted a series of experiments focusing on the textual components that serve as supplementary inputs to the VLM alongside the input chart. The experimental settings are as follows: (1) Title: Use only the title as input to the VLM, without applying the PoT strategy; (2) Stats+Title: Use the PoT strategy to generate a statistics dictionary, combined with the title as input to the VLM; (3) Dict+Title: Use the PoT-generated Python dictionary along with the title as input to the VLM; (4) Dict+StatsT+Title: Replace the PoT strategy with a predefined Python program for generating the statistics dictionary, and use the generated statistics dictionary together with the Python dictionary and title as input to the VLM; (5) Dict+Stats+Title: Use the full set of inputs, including the PoT-generated Python dictionary, the PoT-generated statistics dictionary, and the title as input to the VLM. The experimental results that were evaluated on ROUGE-L and BERTScores are illustrated in Table 4.

VLM Performance Influenced by Input Textual Data While the evaluation results remain influenced by the underlying VLM performance, we observed that in over half of the cases, the combination of the title and Python dictionary outperformed using the title alone. We attribute this to the fact that directly extracted data, despite potential noise, can retain more factual information than purely generated text, potentially steering the model toward more accurate outputs. However, this also highlights the power of using the PoT strategy, as it guides the model to emphasize more on the inaccuracies and noise in the poorly extracted data, while weakening the chart analysis, which negatively harms the overall performance of the model pipeline. In addition, we observed that the PoT-generated statistics dictionaries consistently outperformed the predefined program-generated statistics dictionaries in most cases. This indicates the effectiveness of the PoT strategy, which is better than directly using Python programs to enhance the overall pipeline performance in the chart summarization.

8 Discussion

With empirical results, how a candidate task to represent charts structurally can be an effective auxiliary to the existing chart-to-table task can be sort of answered. While the evaluation of chart-to-table might be more objective in its evaluation, there might be merit to exploring the chart-to-dictionary task for chart understanding. Not only this allows the integration of the chart in a PoT context, but this allows for a more robust representation of the chart, given the increasing complexities of charts in the wild. This work acknowledges that there is an overlap between chart redrawing and this task, but the chart redrawing tends to focus more on the reconstruction of the chart with executable matplotlib code rather than capturing the semantic nuances of the chart elements explored in this work.

The experimentation showed that the summarization task also varied greatly depending on the type of chart the model was captioning. Most models performed well on relatively simpler bar and pie charts, while struggling with more complex charts,

VLM & Textual Data	Pew												VisText							
	Area		Bar		Line		Pie		Scatter		All		Area		Bar		Line		All	
	R-L	BS	R-L	BS	R-L	BS	R-L	BS	R-L	BS	R-L	BS	R-L	BS	R-L	BS	R-L	BS	R-L	BS
deepseek-vl2-tiny																				
Title	**13.57**	**84.78**	**13.57**	**85.49**	**12.26**	**84.83**	**16.98**	**86.96**	**11.99**	**84.22**	**13.33**	**85.34**	**14.65**	**86.87**	**14.44**	**85.69**	**15.68**	**86.89**	**14.79**	**86.30**
Statis+Title	8.22	82.51	8.81	83.73	8.56	83.44	9.21	83.87	8.64	83.37	8.74	83.64	9.90	85.01	8.73	84.23	9.60	84.51	9.26	84.51
Dict+Title	9.51	83.33	6.15	82.04	6.78	82.49	11.80	84.68	7.96	83.44	5.55	82.27	5.37	84.13	3.85	83.26	5.34	84.23	4.23	83.73
Dict+StatisT+Title	8.18	82.47	8.87	83.23	8.95	83.04	10.79	84.06	6.72	81.08	8.92	83.17	10.44	85.15	9.89	84.04	10.71	84.97	10.23	84.56
Dict+Statis+Title	9.16	82.84	10.19	84.33	9.50	83.94	10.79	84.12	8.38	82.90	10.00	84.19	10.88	85.28	9.37	84.22	11.91	85.29	10.38	84.77
internVL-2.5																				
Title	13.80	84.33	13.55	85.02	12.59	84.58	**15.74**	85.59	13.34	84.46	13.38	84.91	10.50	85.17	9.58	84.24	11.28	85.18	10.22	84.71
Statis+Title	13.02	84.91	13.40	85.62	13.15	85.49	12.97	85.62	13.79	85.84	13.32	85.58	12.63	85.81	11.37	84.83	12.90	85.78	12.08	85.32
Dict+Title	**16.15**	85.54	**15.69**	**86.02**	**14.80**	**85.62**	15.73	**86.34**	**15.22**	**85.87**	9.09	85.92	9.58	**86.29**	7.00	85.14	9.69	**86.44**	7.91	85.76
Dict+StatisT+Title	14.74	**85.66**	14.30	85.88	13.68	85.55	15.04	86.00	13.14	84.76	**14.17**	85.79	**13.96**	86.23	**12.49**	**85.19**	**15.10**	**86.44**	**13.52**	**85.78**
Dict+Statis+Title	13.86	85.06	14.17	85.95	13.67	85.58	14.32	86.31	13.28	85.23	14.04	**85.85**	13.43	86.20	11.96	85.18	14.04	86.26	12.85	85.71
qwen2.5-VL-3B																				
Title	14.91	**85.86**	**16.22**	**86.66**	**14.74**	**85.91**	18.38	**87.56**	14.88	**86.12**	**15.88**	**86.49**	**17.78**	**87.30**	**16.39**	**86.19**	**18.98**	**87.31**	**17.40**	**86.76**
Statis+Title	13.64	85.04	14.48	85.98	13.39	85.44	17.57	87.14	13.45	85.42	14.28	85.86	14.19	86.63	13.68	85.68	14.71	86.62	14.06	86.16
Dict+Title	**15.70**	85.76	15.61	86.00	14.35	85.50	**19.53**	87.51	**15.22**	85.36	8.09	85.91	9.38	86.75	6.55	85.71	9.86	86.81	7.46	86.25
Dict+StatisT+Title	13.25	85.28	14.47	85.98	13.03	85.47	18.07	87.02	13.06	85.00	14.19	85.86	14.78	86.79	13.36	85.43	15.96	86.87	14.36	86.15
Dict+Statis+Title	13.91	85.26	14.00	85.83	12.73	85.36	17.16	86.95	13.32	85.15	13.77	85.73	14.15	86.74	13.24	85.60	15.19	86.76	13.94	86.19

Table 4: Ablation study results for different models regarding data used from Pew and VisText datasets evaluated on F1 scores of **R**OUGE-**L** and **B**ERT**S**core scores.

such as multiple line or scatter plots. This indicates that the generalizability of the summarization task may involve some sort of normalization or some way to bridge the gap between the varying levels of complexity presented by the chart. This paper also serves as an empirical springboard for researching specifically on natural language generation for the context of chart summarization, given that most implementations of chart understanding are focused more on question answering.

9 Conclusion

In this work, we conducted a systematic evaluation of the Program-of-Thought (PoT) prompting strategy across currently used lightweight vision-language models under the zero-shot settings on the Pew and VisText benchmarks for the chart summarization task. Our experiments reveal that the efficacy of the PoT varies markedly with model architectures and sizes, corresponding to types of charts, including area, bar, line, pie, and scatter. In this context, the PoT proved to be a competitive alternative to the Direct and MCoT prompting approaches. Beyond prompting strategies, we introduced a novel chart-to-dictionary auxiliary task, demonstrating its promise for capturing robust and semantic nuances in chart understanding, which is also conveniently applicable with the PoT. As charts grow more complex along with the data they represent, there is a need to establish a data structure to evaluate chart-parsing outside the table due to data loss that occurs from the chart to the table.

In the future, we would like to employ complex vision-language models to further explore the impact of PoT strategies on zero-shot chart understanding, particularly in the context of chart summarization. This includes examining how different model architectures and sizes influence the overall chart-to-text pipeline when integrated with PoT strategies. Additionally, we aim to develop more sophisticated PoT approaches capable of generating longer and richer statistical information, thereby enhancing the quality of chart summaries. Since the PoT strategy in this work only extends outputs from short, answer-like responses to relatively concise statistical dictionaries. However, for the chart summarization task, we believe the PoT strategy contains untapped potential to capture factual numeric data by its statistical reasoning capability. Moreover, given the significant influence of the PoT-generated information in model inference, we will also further investigate whether the PoT can contribute to mitigating hallucination errors in the chart summarization process, improving the overall factual accuracy of generated chart summaries.

Limitations

The diverse performance of the PoT strategy across the evaluated models raises several important considerations. The model architecture and size likely play a significant role in determining the effectiveness of different prompting strategies. The models used in this paper were of lightweight VLMs. While effective in the presented lightweight models, the language decoder may have yielded too low conclusive powers on the efficacy of the PoT and CoT prompting methods relative to direct prompting. However, it is seen that the PoT strategy still can offer comparable results to the other prompting methodologies using lightweight VLMs in some cases or for some chart types, which indicates that on higher parameter models, it can be assumed that in the worst case, these different prompting techniques may offer similar results. The research design, comparing three zero-shot prompting methods across four distinct vision-language models and a set of tasks, provides a valuable initial exploration of the PoT's potential on chart summarization with VLMs. Further research can implement few-shot reasoning with examples that can hypothetically increase performance. Additionally, the study focused its experimentation on lightweight VLMs, which might have contributed to the poor results in text generation. Expanding the scope of the study to larger parameter models might lead to more conclusive results. Regarding the evaluation on the summarization task, BLEU, CIDEr, ROUGE, and BERTScore, while attempting to account for fluency and similarity between the target and generated text, also demonstrate that these are not always the effective metrics when comparing generated text as it mostly rely on n-gram overlaps, which ignore factual correctness, semantic similarity, and text informativeness of the captions. There needs to be more automatic metrics focusing on factual consistency, neglecting the exact syntax matching to specifically account for this limitation.

Ethics Statement

To the best of the researchers' knowledge, all datasets used in this study were sourced from publicly available benchmarks. The authors of the benchmark dataset also have obtained the license to distribute the dataset for non-malicious purposes intent which this research has abided by.

References

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Charles Chen, Ruiyi Zhang, Eunyee Koh, Sungchul Kim, Scott Cohen, and Ryan Rossi. 2020a. Figure captioning with relation maps for reasoning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020b. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Raymond Choi, Frank Burns, and Chase Lawrence. 2025. End-to-end chart summarization via visual chain-of-thought in vision-language models. *arXiv preprint arXiv:2502.17589*.

Zhe Cui, Sriram Karthik Badam, M Adil Yalçin, and Niklas Elmqvist. 2019. Datasite: Proactive visual data exploration with computation of insight-based recommendations. *Information Visualization*, 18(2):251–267.

Seniz Demir, Sandra Carberry, and Kathleen F. McCoy. 2012. Summarizing information graphics textually. *Computational Linguistics*, 38(3):527–574.

Massimo Fasciano and Guy Lapalme. 2000. Intentions in the coordinated generation of graphics and text from tabular data. *Knowl. Inf. Syst.*, 2(3):310–339.

Leo Ferres, Gitte Lindgaard, Livia Sumegi, and Bruce Tsuji. 2013. Evaluating a tool for improving accessibility to charts and graphs. *ACM Trans. Comput.-Hum. Interact.*, 20(5).

Leo Ferres, Petro Verkhogliad, Gitte Lindgaard, Louis Boucher, Antoine Chretien, and Martin Lachance. 2007. Improving accessibility to statistical graphs: the igraph-lite system. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 67–74.

Jiayun Fu, Bin B. Zhu, Haidong Zhang, Yayi Zou, Song Ge, Weiwei Cui, Yun Wang, Dongmei Zhang, Xiaojing Ma, and Hai Jin. 2022. Chartstamp: Robust chart embedding for real-world applications. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2786–2795.

Nancy L. Green, Giuseppe Carenini, Stephan Kerpedjiev, Joe Mattis, Johanna D. Moore, and Steven F. Roth. 2004. Autobrief: an experimental system for the automatic generation of briefings in integrated text and information graphics. *Int. J. Hum.-Comput. Stud.*, 61(1):32–70.

Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, and Hanwang Zhang. 2023. Chartllama: A multimodal llm for chart understanding and generation. *Preprint*, arXiv:2311.16483.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Ting-Yao Hsu, C Lee Giles, and Ting-Hao Huang. 2021. SciCap: Generating captions for scientific figures. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3258–3264.

Anwen Hu, Shizhe Chen, and Qin Jin. 2021. Question-controlled text-aware image captioning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3097–3105.

Kung-Hsiang Huang, Hou Pong Chan, Yi R Fung, Haoyi Qiu, Mingyang Zhou, Shafiq Joty, Shih-Fu Chang, and Heng Ji. 2024. From pixels to insights: A survey on automatic chart understanding in the era of large foundation models. *IEEE Transactions on Knowledge and Data Engineering*.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, and 1 others. 2024. Qwen2.5-Coder technical report. *arXiv preprint arXiv:2409.12186*.

Shankar Kantharaj, Rixie Tiffany Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. 2022. Chart-to-text: A large-scale benchmark for chart summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4005–4023.

Hyung-Kwon Ko, Hyeon Jeon, Gwanmo Park, Dae Hyun Kim, Nam Wook Kim, Juho Kim, and Jinwook Seo. 2024. Natural language dataset generation framework for visualizations powered by large language models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *Proceedings of the 40th International Conference on Machine Learning*, pages 18893–18912.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*.

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. 2024. Monkey: Image resolution and text label are important things for large multi-modal models. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

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

Fangyu Liu, Julian Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. 2023a. DePlot: One-shot visual language reasoning by plot-to-table translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10381–10399.

Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Eisenschlos. 2023b. MatCha: Enhancing visual language pretraining with math reasoning and chart derendering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12756–12770.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023c. Improved baselines with visual instruction tuning. *Preprint*, arXiv:2310.03744.

Mengsha Liu, Daoyuan Chen, Yaliang Li, Guian Fang, and Ying Shen. 2024. ChartThinker: A contextual chain-of-thought approach to optimized chart summarization. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3057–3074.

Alan Lundgard and Arvind Satyanarayan. 2022. Accessible visualization via natural language descriptions: A four-level model of semantic content. *IEEE Transactions on Visualization & Computer Graphics (Proc. IEEE VIS)*.

Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. 2023. UniChart: A universal vision-language pretrained model for chart comprehension and reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14662–14684.

Ahmed Masry, Mehrad Shahmohammadi, Md Rizwan Parvez, Enamul Hoque, and Shafiq Joty. 2024.

ChartInstruct: Instruction tuning for chart comprehension and reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10387–10409.

Fanqing Meng, Wenqi Shao, Quanfeng Lu, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. 2024. ChartAssistant: A universal chart multimodal language model via chart-to-table pre-training and multitask instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7775–7803.

Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. 2020. PlotQA: Reasoning over scientific plots. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*.

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):431–467.

Jason Obeid and Enamul Hoque. 2020. 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*, pages 138–147.

OpenAI. 2024. GPT-4o-mini.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Qwen Team. 2025. Qwen2.5-vl.

Ehud Reiter. 2007. An architecture for data-to-text systems. In *Proceedings of the Eleventh European Workshop on Natural Language Generation (ENLG 07)*, pages 97–104.

Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. 2020. TextCaps: A dataset for image captioning with reading comprehension. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II*, pages 742–758.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards VQA models that can read. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hrituraj Singh and Sumit Shekhar. 2020. STL-CQA: Structure-based transformers with localization and encoding for chart question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3275–3284.

Andrea Spreafico and Giuseppe Carenini. 2020. Neural data-driven captioning of time-series line charts. In *Proceedings of the 2020 International Conference on Advanced Visual Interfaces*.

Arjun Srinivasan, Steven M. Drucker, Alex Endert, and John Stasko. 2019. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):672–681.

Benny Tang, Angie Boggust, and Arvind Satyanarayan. 2023. VisText: A benchmark for semantically rich chart captioning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7268–7298.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30:6000–6010.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Yaoting Wang, Shengqiong Wu, Yuecheng Zhang, William Wang, Ziwei Liu, Jiebo Luo, and Hao Fei. 2025. Multimodal chain-of-thought reasoning: A comprehensive survey. *arXiv preprint arXiv:2503.12605*.

Yun Wang, Zhida Sun, Haidong Zhang, Weiwei Cui, Ke Xu, Xiaojuan Ma, and Dongmei Zhang. 2020. Datashot: Automatic generation of fact sheets from tabular data. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):895–905.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*.

Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, Zhenda Xie, Yu Wu, Kai Hu, Jiawei Wang, Yaofeng Sun, Yukun Li, Yishi Piao, Kang Guan, Aixin Liu, and 8 others. 2024. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *Preprint*, arXiv:2412.10302.

Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang.

2024. Tinychart: Efficient chart understanding with program-of-thoughts learning and visual token merging. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1898.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Experiment Set-up

The experiments are conducted with loaded pre-trained models from the vLLM API. As much as possible, the default parameters were used, unless suggested otherwise from official documentation. The temperature is set to 0.2, and the repetition penalty is set to 1.2 across all runs. All experiments are carried out on our machine (CPU: Intel(R) Core(TM) i9-9920X CPU @ 3.50GHz, GPU: 2 NVIDIA RTX3090). Python code generation for producing statistics by the Qwen2.5-Coder-14B-Instruct model is the most computationally costly task, which costs 10-12 hours on 1 GPU.

B More Evaluation Details

B.1 Dataset Analysis

We chose the Pew (Kantharaj et al., 2022) (GPL-3.0 license) and VisText (Tang et al., 2023) (GPL-3.0 license) large-domain English datasets to investigate and evaluate our PoT strategy for generating L2/L3 content in chart summarization, as they provide rich and suitable L2/L3 captions for this task. The VisText is built upon the Statista (Kantharaj et al., 2022) dataset, but with additionally detailed labelled L2/L3 captions. Since the chart labelled in the VisText can have multiple L2/L3 captions, we automatically selected the longest L2/L3 captions in the test set of the Vistext dataset as gold summaries paired to charts for the chart summarization task. The statistics of the Pew and VisText datasets used in this paper are presented in Table 5. In addition, the distribution of topics covered in the Pew and VisText datasets is illustrated in Figure 4.

Statistic	VisText			Pew		
	Simp.	Comp.	All	Simp.	Comp.	All
#Vocab.	3,413	1,995	4,360	3,529	8,342	9,342
Avg.Character	165	152	161	454	522	511
Avg.Token	34	31	33	91	106	104
Avg.Sentence	1.16	0.99	1.11	2.86	3.33	3.26

Table 5: Statistics of datasets by **Simp**le and **Comp**lex complexities of the VisText and Pew test sets.

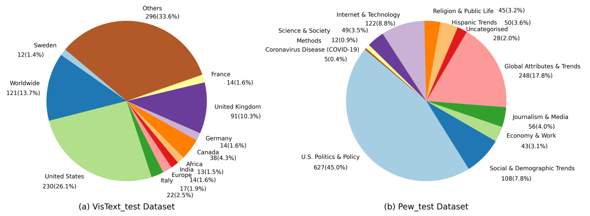


Figure 4: The distributions of topics of VisText and Pew datasets.

B.2 Experiment Implementations

We mainly used Deepseek-VL2 (deepseek-VL2-tiny) (Wu et al., 2024) for testing and our experiments. Additionally, we also tested the following models: InternVL (internVL-2.5-4B) (Chen et al., 2024), LLaVa-NeXT (llava-v1.6-mistral-7b-hf) (Liu et al., 2023c), and Qwen-2.5 (qwen2.5-VL-3B-Instruct) for main and ablation experiments. All experiments were done in Python 3.12 using the vLLM (Kwon et al., 2023) library, with the models being implemented at the zero-shot setting.

B.3 Evaluation Metric Descriptions

To quantitatively measure the performance of our proposed method in chart summarization, we employ two popular automatic evaluation metrics in chart understanding: BLEU (Bilingual Evaluation Understudy) and CIDEr (Consensus-based Image Description Evaluation), in addition to two also well-known automatic evaluation metrics in text summarization: ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2020).

BLEU (Post, 2018) This score calculates the n-gram overlap between the ground-truth summary and the generated summary. It indicates lexical similarity between the generated and ground-truth text, assessing how closely the generated text replicates word sequences that occur in the reference.

CIDEr (Vedantam et al., 2015) This score measures the TFIDF weighted n-gram overlaps between reference and generated text. By weighting n-grams according to their value in a reference summary corpus, CIDEr seeks to more accurately capture the informativeness and relevance of generated descriptions, especially in image and chart captioning tasks.

BLEU and CIDEr are commonly used metrics throughout natural language generation, image captioning, and chart summarization. Together, they capture a more nuanced quantitative measure of model performance in terms of surface similarity and content alignment with reference summaries. While we note that reference-based measures like BLEU and CIDEr do have some limitations, since they can have loose correlation with human preference for aspects of semantic equivalence and factuality, their popularity and ability to provide an initial quantitative score make them effective measures in chart summarization model evaluation. Current research studies on chart summarization have exhaustively employed these metrics as well.

B.4 Additional Ablation Studies

Table 6 and Table 7 present BLEU and CIDEr evaluation results, and ROUGE-1 and ROUGE-L evaluation results, respectively, for various VLMs, tested with different combinations of textual information in our PoT chart summarization pipeline.

C Prompts

C.1 Chart-to-Dictionary Extraction

Similar to the chart-to-table task, this is done in a zero-shot setting. We employ the core concept of PoT to guide the VLM in generating a valid and executable Python dictionary from the input chart.

```
user_prompt = "<img_placeholder> Convert the chart into a python dictionary `chart_dict`. Only consider the chart's data when summarizing."
assistant_ = "```python\n chart_dict ="
```

C.2 Dictionary-to-Statistics with Program of Thoughts

The illustrated prompt content is the same used in VLMs tested in this work, but formatted specifically with each VLM's template.

```
system_prompt = "You are a data analyst. You are given a dictionary that represents a chart called `chart_dict`. \
You need to implement the function `get_summary_statistics(chart_dict)` that takes the dictionary as input and returns a dictionary with the relevant statistics that can be used to summarize the chart. \
Avoid sorting dictionary objects directly and USE ONLY PYTHON BUILT-IN FUNCTIONS. Name the keys of the dictionary to elaborate how it is a descriptive statistic. When writing Python, follow the PEP style guide. \
Return ONLY the code of the function that will run without any errors and can work using `eval()`."

user = "Implement the function `get_summary_statistics` that takes a dictionary as input and returns a dictionary with the relevant statistics that can be used to summarize the chart using only built-in Python functions. Make sure to label the keys of the `summary_dict` to be descriptive The input dictionary is defined as {chart_dict}."

assistant_ = "```python\ndef get_summary_statistics(chart_dict):\n    # Define output dictionary `summary_dict` to store the summary statistics\n"
```

D Case Study

A case study in Figure 7 demonstrates an end-to-end chart-to-text method using the PoT. In this specific instance, the chart-to-dictionary properly captures the appropriate format of how to organize the data, but fundamentally mislabels or misreads the values of which values go to which parties. However, it can be observed that in terms of observing the increasing trend in the time-series data, the dictionary was able to somewhat capture this. The generated PoT is agnostic of the actual values of the functions and is able to correctly identify the relevant keys needed to create summary statistics of total, average, and min and max values. The generated caption captures the general ideas that the chart was able to portray, specifically describing the chart elements of date in the x-axis and anger in the y-axis. While not as verbose as the original text, the generated summary was able to capture the key ideas and trends in the caption.

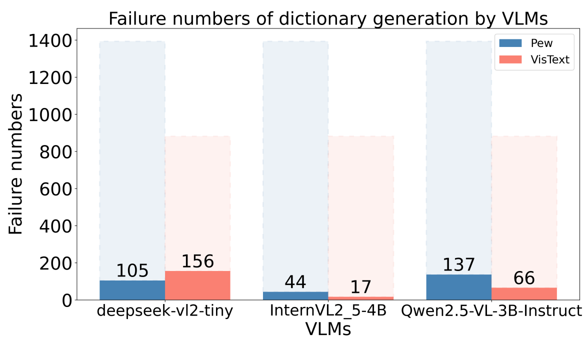


Figure 5: Histogram comparing the numbers of failure cases in the chart data dictionary generation by each VLM on each dataset.

VLM & Textual Data	Pew												VisText							
	Area		Bar		Line		Pie		Scatter		All		Area		Bar		Line		All	
	BLEU	CIDEr	BLEU	CIDEr	BLEU	CIDEr	BLEU	CIDEr	BLEU	CIDEr	BLEU	CIDEr	BLEU	CIDEr	BLEU	CIDEr	BLEU	CIDEr	BLEU	CIDEr
deepseek-vl2-tiny																				
Title	1.9682	0.0427	2.6653	0.0608	1.7169	0.0471	4.5805	0.1391	0.7646	0.0412	2.4676	0.0591	1.8347	0.0920	1.5262	0.0731	2.0429	0.0851	1.7346	0.0824
Statis+Title	0.1254	0.0018	0.2767	0.0127	0.2736	0.0173	0.2496	0.0190	0.2219	0.0005	0.2746	0.0135	0.8102	0.0710	0.3489	0.0523	0.5821	0.0685	0.5603	0.0615
Dict+Title	0.3425	0.0000	0.2343	0.0040	0.1940	0.0055	0.7802	0.0095	0.3621	0.0002	0.1707	0.0025	0.2472	0.0115	0.0853	0.0077	0.2067	0.0124	0.0855	0.0081
Dict+StatisT+Title	0.7589	0.0023	0.4311	0.0170	0.5564	0.0181	0.3408	0.0320	0.3350	0.0279	0.4914	0.0173	0.4408	0.0676	0.7812	0.0568	0.8565	0.0538	0.7502	0.0589
Dict+Statis+Title	0.6960	0.0135	0.6807	0.0236	0.6517	0.0251	0.6614	0.0341	0.3309	0.0011	0.6875	0.0235	0.5583	0.0713	0.4584	0.0737	1.1808	0.0796	0.6914	0.0754
internVL-2.5																				
Title	3.6507	0.0426	3.5832	0.0318	2.7521	0.0296	4.6431	0.1025	2.6224	0.0001	3.4041	0.0328	1.1306	0.0125	0.9387	0.0088	1.3401	0.0212	1.0808	0.0130
Statis+Title	2.8535	0.0713	1.9995	0.0664	1.9136	0.0404	2.0840	0.0907	1.3768	0.0819	1.9896	0.0603	1.1281	0.0246	0.9299	0.0172	1.6892	0.0274	1.1736	0.0212
Dict+Title	3.7973	0.1391	3.1843	0.0650	2.2829	0.0612	2.7083	0.1088	1.6723	0.0569	0.7148	0.0052	0.2476	0.0057	0.0790	0.0023	0.4311	0.0110	0.2141	0.0047
Dict+StatisT+Title	4.1720	0.1772	3.1456	0.0633	2.4598	0.0770	2.7016	0.1286	3.2064	0.0431	3.0008	0.0689	1.7194	0.0555	1.2597	0.0205	2.3460	0.0506	1.5926	0.0371
Dict+Statis+Title	3.6093	0.1211	3.1860	0.0697	2.5661	0.0615	2.9342	0.1188	1.8525	0.0960	3.0319	0.0695	1.4938	0.0326	1.0729	0.0102	1.9497	0.0237	1.3735	0.0192
qwen2.5-VL-3B																				
Title	1.9350	0.0523	3.6251	0.1002	2.5562	0.0643	5.9420	0.1384	2.0714	0.0272	3.3929	0.0905	2.6399	0.1481	2.1772	0.0979	3.1147	0.1519	2.4984	0.1254
Statis+Title	3.3383	0.0409	3.3091	0.0734	2.3678	0.0597	3.8250	0.1662	1.0761	0.0203	3.0906	0.0712	1.6593	0.0780	1.4806	0.0801	2.0928	0.0890	1.6639	0.0826
Dict+Title	2.6846	0.0953	3.1135	0.0693	2.2941	0.0652	3.6053	0.1937	1.5115	0.0629	0.6707	0.0060	0.3687	0.0168	0.0869	0.0090	0.4078	0.0136	0.1515	0.0097
Dict+StatisT+Title	2.4238	0.0222	3.2131	0.0640	2.2744	0.0693	3.4002	0.1018	2.6648	0.0662	2.9969	0.0652	1.7080	0.1080	1.4815	0.0688	2.3149	0.1042	1.6950	0.0883
Dict+Statis+Title	2.6846	0.0953	3.1135	0.0693	2.2941	0.0652	3.6053	0.1937	1.5115	0.0629	2.8237	0.0711	0.3687	0.0168	0.0869	0.0090	0.4078	0.0136	1.5484	0.0678

Table 6: Ablation study results (BLEU / CIDEr) for different models regarding data used from Pew and VisText datasets.

VLM & Textual Data	Pew												VisText							
	Area		Bar		Line		Pie		Scatter		All		Area		Bar		Line		All	
	R-1	R-L	R-1	R-L	R-1	R-L	R-1	R-L	R-1	R-L	R-1	R-L	R-1	R-L	R-1	R-L	R-1	R-L	R-1	R-L
deepseek-vl2-tiny																				
Title	24.62	13.57	25.88	13.57	23.66	12.26	29.17	16.98	24.03	11.99	25.40	13.33	22.37	14.65	21.72	14.44	23.56	15.68	22.33	14.79
Statis+Title	13.45	8.22	13.62	8.81	13.32	8.56	14.03	9.21	15.07	8.64	13.56	8.74	14.08	9.90	11.84	8.73	13.73	9.60	12.91	9.26
Dict+Title	14.82	9.51	8.94	6.15	9.97	6.78	16.71	11.80	13.05	7.96	8.05	5.55	7.35	5.37	4.98	3.85	7.31	5.34	5.56	4.23
Dict+StatisT+Title	15.24	8.18	14.19	8.87	15.34	8.95	16.48	10.79	10.64	6.72	14.53	8.92	15.46	10.44	14.44	9.89	16.12	10.71	15.11	10.23
Dict+Statis+Title	15.86	9.16	16.70	10.19	16.32	9.50	16.81	10.79	13.98	8.38	16.57	10.00	16.48	10.88	13.75	9.37	17.14	11.91	15.29	10.38
internVL-2.5																				
Title	27.44	13.80	28.86	13.55	26.81	12.59	30.08	15.74	27.82	13.34	28.37	13.38	17.17	10.50	16.19	9.58	18.21	11.28	16.92	10.22
Statis+Title	24.41	13.02	25.13	13.40	24.54	13.15	22.33	12.97	26.20	13.79	24.91	13.32	20.59	12.63	18.49	11.37	21.09	12.90	19.68	12.08
Dict+Title	28.78	16.15	28.52	15.69	25.93	14.80	27.67	15.73	27.57	15.22	15.53	9.09	15.87	9.58	10.69	7.00	15.53	9.69	12.43	7.91
Dict+StatisT+Title	26.86	14.74	28.18	14.30	26.66	13.68	26.23	15.04	28.19	13.14	27.73	14.17	22.52	13.96	20.53	12.49	23.43	15.10	21.76	13.52
Dict+Statis+Title	26.64	13.86	28.52	14.17	27.27	13.67	26.11	14.32	27.55	13.28	28.11	14.04	22.24	13.43	20.30	11.96	22.66	14.04	21.38	12.85
qwen2.5-VL-3B																				
Title	24.83	14.91	30.29	16.22	27.70	14.74	32.16	18.38	29.51	14.88	29.62	15.88	26.14	17.78	24.85	16.39	27.12	18.98	25.74	17.40
Statis+Title	24.75	13.64	27.53	14.48	25.53	13.39	29.58	17.57	27.06	13.45	27.06	14.28	22.11	14.19	21.27	13.68	22.70	14.71	21.83	14.06
Dict+Title	26.12	15.70	27.49	15.61	25.49	14.35	30.70	19.53	29.14	15.22	13.50	8.09	14.10	9.38	9.29	6.55	15.00	9.86	10.91	7.46
Dict+StatisT+Title	23.72	13.25	27.44	14.47	25.24	13.03	29.28	18.07	25.69	13.06	26.89	14.19	22.10	14.78	20.74	13.36	23.70	15.96	21.82	14.36
Dict+Statis+Title	25.56	13.91	26.58	14.00	24.62	12.73	28.36	17.16	25.85	13.32	26.13	13.77	22.23	14.15	20.46	13.24	23.20	15.19	21.59	13.94

Table 7: Ablation study results (ROUGE-1 / ROUGE-L) for different models regarding data used on Pew and VisText datasets.

E Failure Case Analysis

E.1 Python Dictionary Generation

In order to keep the desired quality of the statistics in this work, we decided to use InternVL-2.5-4B (Chen et al., 2024) with ChatGPT-4o-mini (OpenAI, 2024) to generate the data dictionary. Figure 5 shows comparisons of failure numbers of the chart data dictionary generation by each VLM, presenting InternVL has the best capability on handling and generating more data dictionaries from the chart data. Since LLaVa is primarily an LLM (LLaMa) with a vision adapter, whereas Deepseek, InternVL, and Qwen are specialized vision-language models with strong visual encoding, we test Deepseek, InternVL, and Qwen on generating the dictionary for chart data on Pew and Vistext datasets, respectively.

But we are aware that most failure cases are due to the wrong structure or format of the Python code, which cannot be recognized or pass the execution tests, and are consequently categorized as failure cases, rather than nonsense or empty outputs. In future work, we will implement a module to refine the Python code into their correct format, ensuring the collection of all valuable data.

E.2 Python Code Generation

Figure 6 presents a comparison between the failure-prone code generated by general-purpose LLMs and the acceptable code produced by code-specialized LLMs, where those models were specif-

```
def get_summary_statistics(chart_dict):
    summaryDictsListedByTypeOfChartAndColorPerCategoryInDictionaryFormat
ForAllCategoriesPresentInInputDictionaryObjectAsKeyValuesPairsInAListWithThe
SameStructureAsOriginalDictionaryObjectButWithoutAnySortingOrDirectManipula
tionOfObjectsInDictionariesWithinItselfToEnsureThatOnlyBuiltInPythonFunctions...
```

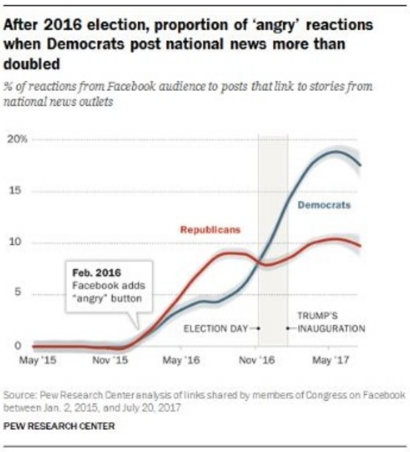
(a) Generated by Qwen2-VL-7B-Instruct

```
def get_summary_statistics(chart_dict):
    summary_dict = {}

    # Extract the title of the chart
    summary_dict['chart_title'] = chart_dict.get('title')

    # Extract the x-axis label
    summary_dict['x_axis_label'] = chart_dict.get('x-axis_label')

     # Extract the y-axis label
    summary_dict['y_axis_label'] = chart_dict.get('y-axis_label')
    return summary_dict
```

(b) Generated by Qwen2.5-Coder-14B

Figure 6: Comparison of failed generated Python code by the general-purpose LLM and the desired generated Python code by the code-specialized LLM.

ically pre-trained and fine-tuned on programming codes, such as Qwen-Coder. With this observation, we chose to use Qwen-2.5-Coder-14B (Hui et al., 2024), which is optimized for generating accurate and efficient code outputs, to ensure the quality of the generated code.



Chart

After Trump won the 2016 presidential election, Democrats in Congress became increasingly likely to share national news on Facebook. Between Jan. 2, 2015, and Nov. 8, 2016 – the day of the presidential election – just 8% of congressional Democrats' Facebook posts included links to national news stories. From Jan. 20 through July 20, 2017, however, that number doubled to 16%.The Center analyzed these reactions and found that, in response to posts from congressional Democrats, the proportion of reactions from the Facebook audience that used the "angry" button more than tripled after the election. From May 1 to Nov. 8, 2016, roughly 5% of reactions to national news links posted by congressional Democrats were "angry" reactions – yet between the election and July 20, 2017, fully 18% of all reactions to Democratic lawmakers' news links were "angry." Reactions to Republican-shared news links remained relatively stable. Between May 1 and Nov. 8, 2016, 9% of reactions to Republican posts were "angry," compared with 10% between Nov. 9, 2016 and July 20, 2017.

Original Text

{"Feb 2016": {"Democrats": 5, "Republicans": 3}, "Nov '15": {"Democrats": 4, "Republicans": 7}, "May '16": {"Democrats": 8, "Republicans": 9}, "ELECTION DAY": {"Democrats": 10, "Republicans": 11}, "Nov '16": {"Democrats": 10, "Republicans": 11}, "Trump's INAUGURATION": {"Democrats": 12, "Republicans": 13}, "May '17": {"Democrats": 15, "Republicans": 16}}

Dictionary Representation

```
def get_summary_statistics(chart_dict):
    summary_dict = {}
    total_democrats = sum(entry['Democrats'] for entry in chart_dict.values())
    total_republicans = sum(entry['Republicans'] for entry in chart_dict.values())
    avg_democrats = total_democrats / len(chart_dict)
    avg_republicans = total_republicans / len(chart_dict)
    max_democrats = max(entry['Democrats'] for entry in chart_dict.values())
    min_democrats = min(entry['Democrats'] for entry in chart_dict.values())
    max_republicans = max(entry['Republicans'] for entry in chart_dict.values())
    min_republicans = min(entry['Republicans'] for entry in chart_dict.values())
    summary_dict['total_democrats'] = total_democrats
    summary_dict['total_republicans'] = total_republicans
    summary_dict['average_democrats'] = avg_democrats
    summary_dict['average_republicans'] = avg_republicans
    summary_dict['max_democrats'] = max_democrats
    summary_dict['min_democrats'] = min_democrats
    summary_dict['max_republicans'] = max_republicans
    summary_dict['min_republicans'] = min_republicans
    return summary_dict
```

PoT

After Trump won the 2016 presidential election, Democrats in Congress became increasingly likely to share national news on Facebook. Between Jan. 2, 2015, and Nov. 8, 2016 – the day of the presidential election – just 8% of congressional Democrats' Facebook posts included links to national news stories. From Jan. 20 through July 20, 2017, however, that number doubled to 16%.The Center analyzed these reactions and found that, in response to posts from congressional Democrats, the proportion of reactions from the Facebook audience that used the "angry" button more than tripled after the election. From May 1 to Nov. 8, 2016, roughly 5% of reactions to national news links posted by congressional Democrats were "angry" reactions – yet between the election and July 20, 2017, fully 18% of all reactions to Democratic lawmakers' news links were "angry." Reactions to Republican-shared news links remained relatively stable. Between May 1 and Nov. 8, 2016, 9% of reactions to Republican posts were "angry," compared with 10% between Nov. 9, 2016 and July 20, 2017.

Original Text

After the 2016 election, there was an increase in angry reactions from Facebook users when Democrats shared stories about national news compared to Republicans. This trend peaked after Trump's inauguration and then began to decline slightly by May 2017. Democrats experienced more anger than Republicans during this period but saw their reaction levels rise significantly following February 2016 when Facebook added its 'angry' button feature for posts linking to national news articles.

Summary Created by PoT using Qwen2.5-VL

Figure 7: Case Study on the generated dictionary, PoT, and generated caption from the experiment trials.