Scaling Down Semantic Leakage: Investigating Associative Bias in Smaller Language Models

Veronika Smilga
University of Tübingen
smilgaveronika@gmail.com

Abstract

Semantic leakage is a phenomenon recently introduced by Gonen et al. (2024). It refers to a situation in which associations learnt from the training data emerge in language model generations in an unexpected and sometimes undesired way. Prior work has focused on leakage in large language models (7B+ parameters). In this study, I use Qwen2.5 model family to explore whether smaller models, ranging from 500M to 7B parameters, demonstrate less semantic leakage due to their limited capacity for capturing complex associations. Building on the previous dataset from Gonen et al. (2024), I introduce a new dataset of color-focused prompts, categorized into specific types of semantic associations, to systematically evaluate the models' performance. Results indicate that smaller models exhibit less semantic leakage overall, although this trend is not strictly linear, with medium-sized models sometimes surpassing larger ones in leaking behavior. The dataset, the model generations, and the evaluation code are publicly available at https://github.com/smilni/semantic_leakage_project.

1 Introduction

Language Models (LMs) have become omnipresent in our everyday lives. However, the way they function is still largely a black-box. Inherently influenced by the training data, LMs demonstrate human-like perception of right and wrong (Schramowski et al., 2021), while also exhibiting a variety of undesirable social biases (Bolukbasi et al., 2016; Lucy and Bamman, 2021). Navigli et al. (2023) argue that the source of such biases are associations learnt by the model in the course of its training.

A broader example of learnt associations surfacing in an unexpected and unsuitable way is **semantic leakage** (Gonen et al., 2024). The association learnt by the model might make a concept from a prompt influence the model generation out of proportion. That results in bizarre outputs, such as *"The dinner was served on pink plates. Today's dish was... **rose petal soup**"*[1]. Another interesting case of semantic leakage is the leakage of literal word-by-word meaning of idioms and proper names, such as *"Ivory is a student. Her favorite color is... **white**"*[2].

Semantic leakage in LMs is closely related to problems in word-to-concept mapping in image generation. With prompts containing several objects, the attributes of the objects may leak into each other, get interchanged, or one of the objects may be omitted from the image altogether (Feng et al., 2023; Dahary et al., 2024). Conversely, the presence of one ambiguous word in a prompt may lead to the model producing a picture of several entities corresponding to each of the word's meanings (Rassin et al., 2022).

While the above-described problems with word-to-concept mapping problems lead to errors in image generation, one may argue that semantic leakage in text generation, on the opposite, helps in maintaining the coherence of the text. Both semantic leakage and bias in machine learning models seem to be rooted in undesirable associations learnt during training. Unlike inherent model bias, semantic leakage does not necessarily result in incorrect or harmful generations. However, as pointed out by Gonen et al. (2024), semantic leakage is a common pattern that characterises language models across different sizes and architectures, large and small, instruction-tuned and general-purpose ones. It would thus be interesting to investigate the phenomenon further to understand how models internalize and use associations.

Gonen et al. (2024) analyse semantic leakage in various categories with strong semantic association component, including colors, food, animals, songs,

---

[1]Generated by Qwen2.5-3B-Instruct
[2]Generated by Qwen2.5-1.5B-Instruct

and occupations. They test several Large LMs, including models of Llama-2 7B+ and LLama-3 8B+ families as well as OpenAI's GPT3.5, GPT4 and GPT4o.

In this project, I further investigate the phenomenon of semantic leakage. Building on Gonen et al. (2024) who examined models of 7 billion parameters and larger, I choose to focus on smaller LMs, from 500 million to 7 billion parameters. The goal is to determine whether the model's size affects its proneness to semantic leakage. I hypothesize that smaller models, due to their size, may be unable to capture the semantic associations that result in semantic leakage. Thus, I am expecting smaller LMs to exhibit less leaking behavior as compared to larger ones.

Additionally, I build a new set of prompts focusing on the category of color and test the models both on the original dataset by Gonen et al. (2024) and the color-focused one. Color-related prompts are divided into three categories: 1) prompts with a mention of a color, with a non-color-related concept to be generated; 2) prompts with a mention of a color, with another color to be generated; 3) prompts with names or set expressions mentioning a color, with another color to be generated[3]. The goal is to test whether the models exhibit leakage more for any of the categories. I hypothesize that prompts containing a color with another color to be generated, i.e. prompts of type 2) and type 3), leak more as models may take short-cuts by repeating previously mentioned colors.

2 Methods

2.1 Models

Qwen2.5 (Team, 2024) is a series of large language models sized from 500 million to 72 billion parameters, both general-purpose and instruction-fine-tuned. Qwen2.5 is the most capable state-of-the-art series of models with open weights and smaller counterparts available as of today[4]. I test all Qwen2.5-Instruct models up to 7B parameters: Qwen2.5-0.5B-Instruct, Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct, Qwen2.5-7B-

Instruct-GPTQ-Int4[5]. Only Instruct models are tested as, according to Gonen et al. (2024), instruction-finetuned models exhibit more leaking behavior than their general-purpose counterparts. In addition to that, I found it hard to get meaningful generations from smaller non-instruction-finetuned Qwen models.

All models were accessed using Huggingface[6]. Inference was run using Kaggle notebooks[7] on a single P100 GPU. Temperature was set to an intermediate value of 0.5 to ensure balance between exploring diverse token probabilities and preserving high-probability sequences. Maximum number of tokens to be generated was set to 10. Additionally, generated text was capped after the first full stop, exclamation or question mark, following the original paper.

2.2 Evaluation metrics

To evaluate how much a model exhibit semantic leakage, I use Mean Leak-Rate suggested by Gonen et al. (2024). This metric operates by comparing similarity of a concept that is supposed to trigger semantic leakage, a control generation for a prompt without the concept and a test generation for a prompt containing the concept. Let us examine the example from 1, *"Ivory is a student. Her favorite color is... white"*. The leakage-triggering concept is *Ivory*, a name that is likely to be associated with color white due to the original meaning of the word *ivory*. The test prompt leading to test generation is *"Ivory is a student. Her favorite color is"*, and the control prompt leading to control generation is *"Her favorite color is"*. Test generation *white* is an example of leaking behavior.

Leak-Rate is the percentage of instances when the similarity of the concept and the test generation is higher than that of the concept and the control generation. Leak-Rate for one prompt is calculated as follows:

$$\text{Leak-Rate} = \begin{cases} 100 & \text{if } sim(\text{concept, test}) > sim(\text{concept, control}) \\ 0 & \text{if } sim(\text{concept, test}) < sim(\text{concept, control}) \\ 50 & \text{if } sim(\text{concept, test}) = sim(\text{concept, control}) \end{cases}$$

(Gonen et al. 2024)

To obtain the final Mean Leak-Rate of a model, the Leak-Rate is then averaged across all test

---

[3]Refer to Subsection 2.3 for example prompts of each category.

[4]As of October 1st, 2024, according to Chatbot Arena LLM Leaderboard (Chiang et al., 2024), among models with open weights, Qwen2.5-72b-Instruct is only surpassed by Meta-Llama-3.1-405b-Instruct. However, Llama-3.1 model family only features models of 8B, 70B, and 405B parameters, while I aim to evaluate models of 7B and smaller.

[5]Dut to computational constraints, I had to use GPTQ-quantised (Frantar et al., 2023) version of Qwen2.5-7B-Instruct that would fit into a single P100 GPU.

[6]https://huggingface.com

[7]https://www.kaggle.com

Cat.	Prompt type	Generation type	Example prompt & generation
1	Color	Non-color-related item	He likes **red**. He works as a... **firefighter**
2	Color	Color	**White** clouds floated in the summer sky. The fence was painted... **white and weathered**
3	Color-related name or set expression	Color	Mary saw the world **through rose-colored glasses**. She was wearing a T-shirt colored... **pink**

Table 1: Dataset categories. Color-related concepts in prompt are in black bold. Color-related concepts in model generation are in green bold. All example generations were obtained from Qwen2.5-7B-Instruct.

Dataset	Model	BERT-score	SentenceBERT
Various semantic categories (Gonen et al., 2024)	0.5B-Instruct	69.27	77.52
	1.5B-Instruct	75.23	79.36
	3B-Instruct	**83.03**	73.85
	7B-Instruct-GPTQ-Int4	74.77	**80.64**
Color-related (the new dataset)	0.5B-Instruct	60.68	61.23
	1.5B-Instruct	71.77	69.58
	3B-Instruct	**77.41**	**80.33**
	7B-Instruct-GPTQ-Int4	73.41	66.39

Table 2: Mean Leak-Rate for original and new color-related dataset, Qwen2.5 model family. The highest score for each dataset and score type across the model sizes is in bold.

prompts:

$$\text{Mean Leak-Rate} = \frac{1}{N} \sum_{i=1}^{N} \text{Leak-Rate\_i} \quad (1)$$

where Leak-Rate_i is the Leak-Rate for prompt i and N is the total number of prompts.

To compute similarity between concepts and generations, I use BERT-score (Zhang et al., 2020) with distilbert-base-uncased[8] as the backbone model and SentenceBERT (Reimers and Gurevych, 2019) with all-MiniLM-L6-v2[9] as the backbone model. Since the original paper's (Gonen et al., 2024) code is not yet available, I implement the calculations myself, following the description provided by the authors.

2.3 Datasets

As a starting point, I evaluate the models' proneness to semantic leakage on the initial dataset of 109 prompts related to various semantic categories from Gonen et al. (2024). In order to calculate Leak-Rate on the original dataset, I manually iden-

tify the concept and the corresponding control item for each test prompt.

Additionally, I build a new extended dataset of prompts related to colors. This dataset consists of three parts: 1) prompts with color as the concept and an expected non-color-related generation; 2) prompts with color as the concept and an expected color-related generation; 3) prompts with color-related name or set expression as the concept and an expected color-related generation. See Table 1 for an example of a prompt and a generation for each category.

To construct prompts for categories 1) and 2), I handcraft 20 templates (10 for each category), such as *"The flowerpot is {INSERT}. The flower is a ..."*. I make sure that in each case any color can be inserted to replace the placeholder *{INSERT}*, i.e. none of the templates are tied to specific colors. To measure semantic leakage as described in Subsection 2.2, each template has a corresponding control prompt, e.g. *"The flower is a ..."*.

Each template is then completed using 11 basic color terms, resulting in 220 prompts. Following Berlin and Kay (1991), I define basic color terms for English as black, white, red, green, yellow, blue, brown, orange, pink, purple, and gray. To further diversify the dataset, I also include two

Cat.	Model	BERT-score	SentenceBERT
1) Color in prompt, other concept expected in generation	0.5B-Instruct	65.03	**70.61**
	1.5B-Instruct	**73.18**	**72.79**
	3B-Instruct	70.45	73.76
	7B-Instruct-GPTQ-Int4	**88.45**	**70.3**
2) Color in prompt, color expected in generation	0.5B-Instruct	56.33	51.85
	1.5B-Instruct	70.36	66.36
	3B-Instruct	**84.36**	**86.91**
	7B-Instruct-GPTQ-Int4	58.36	62.48
3) Name or set expression in prompt, color expected in generation	0.5B-Instruct	**71.22**	**70.61**
	1.5B-Instruct	66.33	61.02
	3B-Instruct	57.76	72.45
	7B-Instruct-GPTQ-Int4	60.82	64.9

Table 3: Mean Leak-Rate for different categories of color-related prompts, Qwen2.5 model family. The highest score for each model size across all categories is in bold.

more specific shades or variations for each basic color, resulting in the final dataset of 660 prompts for categories 1) and 2). You may find the full list of colors used to complete the templates in Appendix A.

To construct prompts for category 3), I handcraft 4 templates, such as *"{INSERT} is a student. Her favorite color is ..."* for names and *"{INSERT} He was wearing a T-shirt colored ..."* for set expressions. Same as for other categories, each template has a corresponding control prompt.

When constructing prompts for category 3), simple slot-filling was not always sufficient. Prompts featuring set expressions were created manually to ensure that the result was coherent and the set expression was contextually relevant. The final dataset contains 60 prompts of category 3), each featuring a unique color-rated proper name or expression. You may find the full list of color-related names and set expression used to complete the templates in Appendix B.

3   Results

Table 2 presents the resulting Leak-Rate scores calculated with the use of BERT-score and Sentence-BERT for the four models of Qwen2.5-Instruct family for the original and the new dataset. For all models and datasets, the Leak-Rate exceeds 50%, meaning that test generations show higher semantic similarity to the corresponding concepts in comparison with control generations. A 50% Leak-Rate would indicate random similarity and no leakage, while values above 50% confirm that semantic leakage is present in the models' output.

Models of all sizes demonstrate consistently high Leak-Rate on the original dataset from Gonen et al. (2024). Interestingly, compared to Llama and OpenAI models evaluated in Gonen et al. (2024), Qwen2.5 model family appears to be even more prone to leakage. Qwen2.5-3B-Instruct achieves the highest BERT-score-based Mean Leak-Rate of 83.03, surpassing the highest BERT-score-based Mean Leak-Rate of 78.1 reported for Llama-3-8B-Instruct by Gonen et al. (2024). Similarly, Qwen2.5-7B-Instruct-GPTQ-Int4 demonstrates the highest SentenceBERT-based Mean Leak-Rate of 80.64, compared to the highest score of 71.7 reported for Llama-2-7B-Chat in the original paper.

Larger models seem to demonstrate higher Leak-Rate, with Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct-GPTQ-Int4 having the highest BERT-score-based and SentenceBERT-based Mean Leak-Rate score for both original dataset and newly constructed color-related dataset. Interestingly, the smallest model, Qwen2.5-0.5B-Instruct, seems to demonstrate the least leaking behavior on both datasets in terms of both BERT-score-based and SentenceBERT-based Mean Leak-Rate.

The hypothesis that smaller language models leak less information than larger ones appears to hold, at least partially. Indeed, Qwen2.5-0.5B-Instruct is the smallest model of the family, and its Mean-Leak Rate is consistently the lowest or next to the lowest. However, Qwen2.5-7B-Instruct-GPTQ-Int4 is the largest model of the family included in my analysis, but in three out of four cases it demonstrates lower Mean Leak-Rate than Qwen2.5-3B-Instruct. I would generalize that

within the Qwen2.5 family models under 7 billion parameters tend to exhibit more leakage as their size increases, but this relationship is not strictly linear – in most cases, the highest Mean Leak-Rate score is demonstrated by the second-largest model.

Table 3 presents the BERT-score-based and SentenceBERT-based Leak-Rate scores for each of the categories of the color-related dataset. Models of different sizes demonstrate distinct behaviors in terms of the degree of semantic leakage across different categories. However, after averaging the Mean Leak-Rate for each category, it is evident that in general models tend to leak more for prompts belonging to category 1), with mean BERT-score-based LR of 74.28 and SentenceBERT-based LR of 71.87, as compared to category 2), with mean BERT-score-based LR of 67.35 and SentenceBERT-based LR of 66.90, and category 3), with mean BERT-score-based LR of 64.03 and SentenceBERT-based LR of 67.25.

The hypothesis that prompts of categories 2) and 3), i.e. containing a color and leading to generation of another color, would lead to more leaking behavior, does not generally hold true. It is only true for Qwen2.5-3B-Instruct because this model tends to repeat parts of the prompt in the generation, as I will further demonstrate in Section 4. Contexts like those in 2) and 3) make concept repetition particularly convenient as the same kind of concept is featured in the prompt and expected to be generated. However, in general, models, including Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct-GPTQ-Int4 tend to leak more for prompts of category 1), i.e. prompts with a concept other than a color to be generated.

4 Discussion

It is important to note that high Mean Leak-Rate of Qwen2.5-3B-Instruct is mostly explained by the fact that this model tended to repeat the concept mentioned in the prompt in its generations, e.g. *"The cake was topped with tangerine orange icing. The candles were colored... **tangerine orange**"*. None of the other models, smaller or larger, exhibit repetitions to such degree. All models were run with the same hyperparameters, so I find this tendency hard to explain. However, it is important to take this peculiarity of Qwen2.5-3B-Instruct into account when drawing conclusion about the leaking behaviour of the models.

The lower Mean Leak-Rate of Qwen2.5-0.5B-Instruct and Qwen2.5-1.5B-Instruct may be explained by the fact that these models tend to generate the same continuations to the prompts regardless of the preceding context. For example, in *"The house was painted charcoal black / emerald green / chestnut brown / coral orange. She saw a... **tree in the yard**"*[10], the generation is the same for different colors in prompt. In other words, the immediate context in the end of the prompt seems to affect the model more than the potential leakage triggering concept.

That does not happen with larger models – both Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct-GPTQ-Int4 seem to be much more sensitive to the context, with few to none repeated generations for different prompt variations. For example, for the above-mentioned template, there each color triggers a different generation: *"The house was painted charcoal black / emerald green / chestnut brown / coral orange. She saw a... **ghostly figure / shimmering hue / squirrel on the roof / sudden surprise**"*[11]

It seems that semantic leakage does indeed stem from the associations learnt by the model during training. In cases like the one described above, semantic leakage adds to the lexical richness and diversity of the generated text. Small models, such as Qwen2.5-0.5B-Instruct, exhibit less leaking behavior, but at the same time provide less diverse and less context-aware generations. In other words, it is important to understand that semantic leakage is not inherently negative but rather a byproduct of the associative knowledge learnt by the model. While undesirable in certain scenarios, such as those requiring unbiased or contextually accurate outputs, semantic leakage can make a positive contribution in other cases.

One of the limitations of the current study is the way semantic leakage is measured. In some cases, the repetition of the concept from the prompt in the generation is justified as it is used to refer to a previously mentioned concept, e.g. *"The house was painted green. She saw a... **green house**"*[12]. However, generations like that are definitely semantically related to the concept of *green* from the prompt and thus get classified as instances of semantic leakage, unjustly increasing the Mean Leak-Rate of the model.

---

[10]Generated by Qwen2.5-0.5B-Instruct
[11]Generated by Qwen2.5-7B-Instruct
[12]Generated by Qwen2.5-7B-Instruct-GPTQ-Int4

Another limitation of the current study is the imbalance of categories in the new color-related dataset, with 330 prompts of category 1), 330 prompts of category 2), and only 60 prompts of category 3). This situation is explained by the fact that prompts of categories 1) and 2) were produced semi-automatically, by slot-filling colors in pre-written templates, while prompts of category 3) were crafted manually in an attempt to preserve coherence in prompts featuring set expressions. However, such imbalance makes our conclusions about the differences in semantic leakage for each of the categories less reliable.

Possible directions for future work include addressing the category imbalance. The smallest category 3) can be expanded by using the same names and set expressions in more than one prompt. Additionally, one may want to test larger models of the same family, Qwen2.5-14B-Instruct, Qwen2.5-32B-Instruct, and Qwen2.5-72B-Instruct, to determine whether the tendency of larger models to leak more holds for 14B+ models. Finally, it may be interesting to investigate the ways of detecting and mitigating undesirable leaking behavior in language models.

References

Brent Berlin and Paul Kay. 1991. *Basic color terms: Their universality and evolution*. Univ of California Press.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating llms by human preference.

Omer Dahary, Or Patashnik, Kfir Aberman, and Daniel Cohen-Or. 2024. Be yourself: Bounded attention for multi-subject text-to-image generation.

Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. 2023. Training-free structured diffusion guidance for compositional text-to-image synthesis.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. Gptq: Accurate post-training quantization for generative pre-trained transformers.

Hila Gonen, Terra Blevins, Alisa Liu, Luke Zettlemoyer, and Noah A Smith. 2024. Does liking yellow imply driving a school bus? semantic leakage in language models. *arXiv preprint arXiv:2408.06518*.

Li Lucy and David Bamman. 2021. Gender and representation bias in GPT-3 generated stories. In *Proceedings of the Third Workshop on Narrative Understanding*, pages 48–55, Virtual. Association for Computational Linguistics.

Roberto Navigli, Simone Conia, and Björn Ross. 2023. Biases in large language models: Origins, inventory, and discussion. *J. Data and Information Quality*, 15(2).

Royi Rassin, Shauli Ravfogel, and Yoav Goldberg. 2022. Dalle-2 is seeing double: Flaws in word-to-concept mapping in text2image models.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Patrick Schramowski, Cigdem Turan, Nico Andersen, Constantin A. Rothkopf, and Kristian Kersting. 2021. Language models have a moral dimension. *CoRR*, abs/2103.11790.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

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

A Basic colors and their variations, dataset categories a) and b)

Basic color	Variations
yellow	golden yellow, lemon yellow
red	crimson red, scarlet red
purple	violet purple, lavender purple
green	emerald green, olive green
blue	navy blue, cobalt blue
pink	fuchsia pink, blush pink
black	charcoal black, jet black
white	ivory white, snow white
brown	chestnut brown, mahogany brown
orange	coral orange, tangerine orange
gray	slate gray, ash gray

B Color-related names and set expressions, dataset category c)

Type	Example
Names	Rose, Amber, Ruby, Scarlett, Violet, Jade, Hazel, Indigo, Ivory, Coral, Sienna, Olive, Ebony, Gray, Pearl
Toponyms	Yellowknife (Canada), Redmond (Washington), Purple Springs (Alberta), Greenwich (UK), Bluefield (West Virginia), Roseland (Chicago), Blackfoot (Idaho), White Plains (New York), Brownsville (Texas), Orange (France), Grayling (Michigan)
Set expressions	to be a black sheep, to be tickled pink, to be in the red, to tell a white lie, to come out of the blue, to see a golden opportunity, to be in a gray area, to get the green light, to paint the town red, to have a silver tongue, to have a rosy outlook, to be as gray as the cloudy sky, to be in a black mood, to think in black and white, to have white-hot anger, to feel like a green rookie, to have rosy cheeks, to be in a brown study, to see the world through rose-colored glasses