Towards Understanding In-Context Learning with Contrastive Demonstrations and Saliency Maps

FUXIAO LIU*, PAIHENG XU*, ZONGXIA LI*, YUE FENG, HYEMI SONG
UNIVERSITY OF MARYLAND, COLLEGE PARK

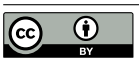
1 ABSTRACT

We investigate the role of various demonstration components in the in-context learning (ICL) performance of large language models (LLMs). Specifically, we explore the impacts of ground-truth labels, input distribution, and complementary explanations, particularly when these are altered or perturbed. We build on previous work, which offers mixed findings on how these elements influence ICL. To probe these questions, we employ explainable NLP (XNLP) methods and utilize saliency maps of contrastive demonstrations for both qualitative and quantitative analysis. Our findings reveal that flipping ground-truth labels significantly affects the saliency, though it's more noticeable in larger LLMs. Our analysis of the input distribution at a granular level reveals that changing sentiment-indicative terms in a sentiment analysis task to neutral ones does not have as substantial an impact as altering ground-truth labels. Finally, we find that the effectiveness of complementary explanations in boosting ICL performance is task-dependent, with limited benefits seen in sentiment analysis tasks compared to symbolic reasoning tasks. These insights are critical for understanding the functionality of LLMs and guiding the development of effective demonstrations, which is increasingly relevant in light of the growing use of LLMs in applications such as ChatGPT.

2 INTRODUCTION

Large language models (LLMs) show significant ability of in-context learning (ICL) for many NLP tasks [Brown et al. 2020; Yin et al. 2023; Zhao et al. 2023]. ICL only requires a few input-label pairs for demonstrations and does not require fine-tuning on the model parameters. However, how each part of the demonstrations used in ICL drives the prediction remains an open research question. Previous works have mixed findings. For examples, although one might assume that ground-truth labels would have a similar impact on ICL as they do on supervised learning, Min et al. [2022] finds that the ground truth input-label correspondence has little impact on the performance of end tasks. However, Zhao et al. [2021] suggests that the example ordering has a strong impact. More recently, Wei et al. [2023] find that only LLMs with larger scales can learn the flipped input-label mapping.

In this work, we use XNLP methods to understand which part of the demonstration contributes to the predictions more. We are interested in the impact of contrastive input-label demonstration pairs built in different ways, i.e., flipping the labels, changing the input, and adding complementary explanations as shown in Fig. 1. We then contrast the saliency maps of these contrastive demonstrations via qualitative and quantitative analysis. Prior works [Brown et al. 2020; Liu et al. 2023a, 2020; Min et al. 2022; Wei et al. 2023] show LLMs in relatively small scale, such as all GPT-3 models [Brown et al. 2020] (based on categorization in [Wei et al. 2023]), cannot override prior knowledge from pretraining with demonstrations presented in-context, which means LLMs do not flip their predictions when the ground-truth labels are flipped in the demonstrations [Min et al. 2022]. However, Wei et al. [2023] show larger models like InstructGPT [Ouyang et al. 2022] (specifically the `text-davinci-002` checkpoint) and PaLM-540B [Chowdhery et al. 2022] have the



emergent ability to override prior knowledge in the same setting. We partly reproduce the results from previous work [Min et al. 2022; Wei et al. 2023] on a sentiment classification task and find that the ground-truth labels in the demonstration are less salient after label flipping.

Meanwhile, as the other important part of the demonstrations, the effect of input distribution is understudied. Min et al. [2022] change the whole input to random words and Wei et al. [2023] do no investigate input distribution at all. Therefore, we investigate the impact of input distribution at a fine-grained level, where we edit the input text's different components in correspondence to task-specific purposes. In the case of sentiment analysis, we change the sentiment-indicative terms in the input text of demonstrations to sentiment-neutral ones. We find that such input perturbation (neutralization) does not have as large impact as changing ground-truth label do. We suspect the models rely on pretrained knowledge to make fairly good predictions because the averaged importance scores for neutralized terms are smaller than the ones of original sentiment-indicative terms.

Additionally, we find that complementary explanations do not necessarily benefit sentiment analysis task as they do for symbolic reasoning tasks as shown in [Ye et al. 2022], even though the saliency maps suggest the explanations tokens are as salient as the original input tokens. This suggests that we need to carefully generate complementary explanations and evaluate whether the target task would benefit from them when trying to boost ICL performance with such technique.

We hope the findings of this study can help researchers better understand the mechanism of LLMs and provide insights for practitioners when curating the demonstrations. Especially with the recent popularity of ChatGPT, we hope this study can help people from various domains have a better user experience with LLMs. The code for this study is available at https://github.com/paihengxu/XICL.

3 BACKGROUND

3.1 Understanding ICL

Large language models (LLMs) show significant ability of in-context learning (ICL) for many NLP tasks [Brown et al. 2020; Liu et al. 2023b; Yin et al. 2023; Zhao et al. 2023]. Min et al. [2022] show that presenting random ground truth labels in the demonstrations does not substantially affect performance. They also change other parts of the demonstrations (e.g., label space, distribution of the input text and overall sequence format) and find these factors are the key drivers for the end task performance. Wei et al. [2023] concentrates on labels by comparing LMs across different size scales with two variants that have flipped labels or semantically-unrelated labels. They find that only large LMs can flip the predictions to follow flipped demonstrations. Akyürek et al. [2022] and Xie et al. [2021] try to understand in-context learning by training transformer-based in-context learners on small-scale synthetic datasets.

3.2 Saliency Maps

3.2.1 Gradient-based Methods. For models with parameter access, we can estimate the importance of an input token using derivative of output w.r.t that token. The most basic method assigns importance by the gradient [Simonyan et al. 2014]. However, it suffers from some known issues such as sensitivity to slight perturbations, saturated outputs, and discontinuous gradient. SmoothGrad [Smilkov et al. 2017] reduces the noise in the importance scores by adding Gaussian noise to the original input. Integrated Gradients (IG) [Sundararajan et al. 2017] computes a line integral of the vanilla saliency from a baseline point to the input in the feature space.

3.2.2 Perturbation-based Methods. An alternative approach to generating saliency maps using input perturbations can be applied to black-box models. Instead, the process involves systematically altering the input data (i.e., words, phrases, and sentences) and observing the changes in the model's

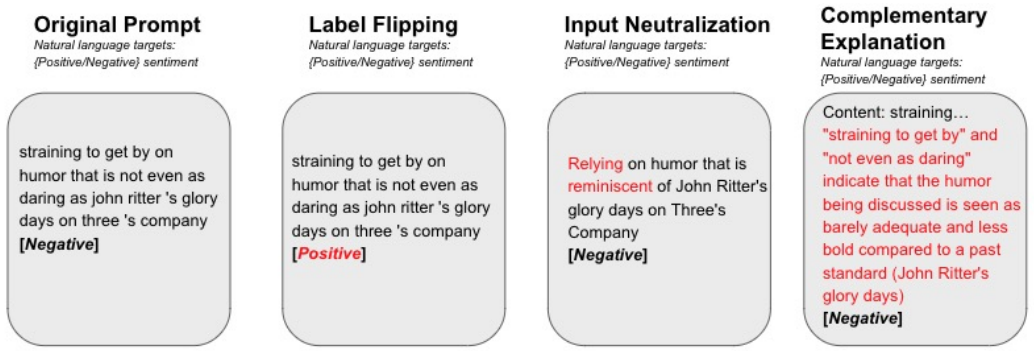


Fig. 1. An overview of three ways to build contrastive demonstrations - flipping labels, perturbing (neutralizing) input, and adding complementary explanations. The contrastive parts are colored in red.

output. We plan to start with the standard method that falls into this category, LIME [Ribeiro et al. 2016]. The process involves creating perturbed versions of an input instance, passing them through the model, training a local linear model on the perturbed inputs and their corresponding predictions, and extracting feature importances from the local model.

4 APPROACH

Despite previous efforts on understanding ICL [Akyürek et al. 2022; Min et al. 2022; Wei et al. 2023; Xie et al. 2021], we are the first attempt to understand ICL using XNLP techniques to the best of our knowledge. We build contrastive demonstrations in various ways and contrast the saliency maps of these contrastive demonstrations to the ones of the original demonstrations to better understand ICL.

We adopt the following three methods to build contrastive demonstrations: flipping labels, perturbing (neutralizing) input, and adding complementary explanations, as shown in Fig. 1. We follow [Min et al. 2022; Wei et al. 2023] to flip the labels in the demonstration. Min et al. [2022] changed input text distribution to random English words, we focus on a task-specific perturbation in this study. Since we use sentiment analysis as our task and adjectives are strong and sometimes causal indicators of the prediction in this task, we neutralize adjectives in the demonstrations. Ye et al. [2022] show that adding complementary explanations benefits ICL. We want to investigate how important are these explanations using saliency map methods. We hope that comparing the saliency maps of these contrastive prompts with the ones of the original prompt would give us insights into how different parts of the demonstrations contribute to ICL predictions.

5 EXPERIMENTAL SET-UP

Dataset. We choose **SST-2** [Socher et al. 2013], a sentiment analysis task, as our baseline task to explain ICL paradigm. Due to budget limitations and to follow [Min et al. 2022; Wei et al. 2023], we randomly sampled 288 examples that are not shorter than 20 tokens from the SST-2 training set as the test set. Additionally, we randomly sample 20 examples for generating saliency maps.

5.1 Demonstration Selection

Other than the 288 examples we picked from the training set, we also hand picked 4 example demonstrations to test language models' in-context-learning ability. We picked two examples with positive labels and two examples with negative labels to have an even distribution of classes in

our demonstrations. The demonstrations have significant word indicators for positiveness and negativeness. As in Fig. 2, shows full demonstrations under four conditions: original demonstrations, label flipping, input neutralization, and adding explanation for each demonstration.

Label Flipping. We flip the binary label for each of the prompt and use them as the demonstrations during testing.

Input Neutralization. For each of the demonstration, we prompted GPT-4 with its review, and asked it to replace indicative words or phrases (positive or negative phrases) that could lead to the labels for the prompts with neutral words and phrases. After GPT-4 generates a perturbed version of the original review, we manually examine the validity of the perturbed prompts and make sure changes from the original prompts are minimal.

Complementary Explanation. We add a complementary explanation for each of the review demonstration as shown in Fig 2d. The explanations are generated by prompting GPT-4 with 'Can you give an explanation to why (REVIEW) is labled positive/negative?'. After each explanation was generated, we manually rephrase the explanation to make it shorter and more concise.

5.2 Baseline LMs and Metric

We first evaluate accuracy of the following models on the sampled SST-2 dataset.

Fine-tuned BERT. We used BERT fine-tuned on SST-2 dataset to get a supervised accuracy of a language model[1] for a better comparison on other models' accuracy on the same dataset without fine-tuning.

ChatGPT-3.5-turbo. We used the openAI gpt-3.5-turbo, with maximum number response token length set to 4096 tokens, temperature set to 0 (with no randomness of the outputs), and top_n set to 1, which always gives the top 1 prediction of the model.

Instruct-GPT. We used `text-davinci-002`, a member of the Instruct-GPT family, with temperature 0, maximum tokens generated equal to 50 (to keep cost in control), and top_n also set to 1.

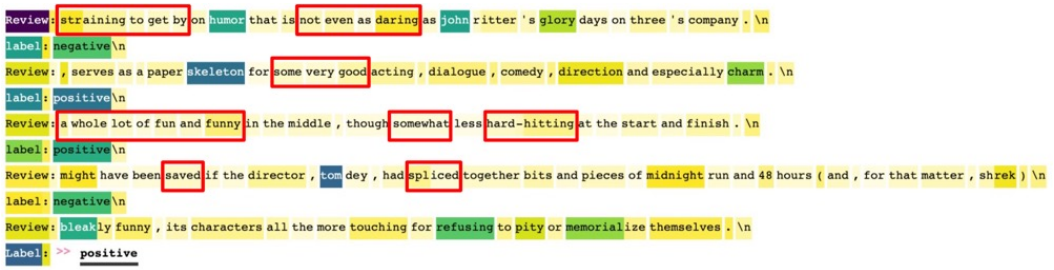
GPT-2. We select GPT-2 [2] with 124M parameters, the smallest version of GPT-2. The GPT-2 model does not sample when generating the output (i.e., temperature set to 0). We are not able to run larger GPT-2 models like GPT-medium, GPT-large, and GPT-XL on a single GPU available at hand.

Metric. We use the accuracy to evaluate sentiment classification. We also use T-test to verify our hypothesis on the saliency map patterns for the three contrastive demonstrations.

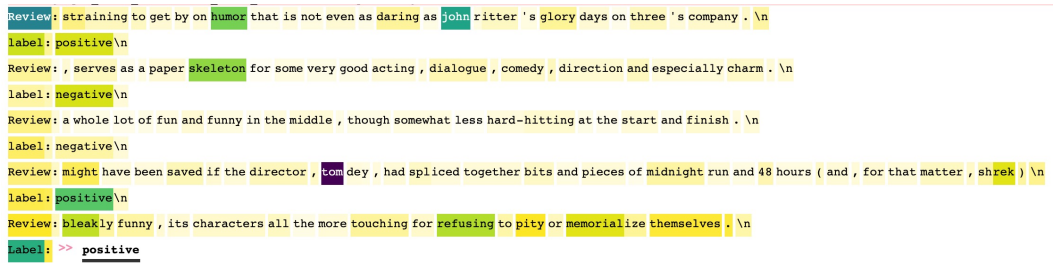
5.3 Saliency Map Methods

For gradient-based method, we use IG [Sundararajan et al. 2017] for models with parameter access (i.e., GPT-2) with the implementation from Ecco [Alammar 2021]. For black-box models (i.e., `text-davinci-002` from Instruct-GPT family), we use LIME[3] to explain the Instruct-GPT classifier. We use the LimeTextExplainer to explain the instance, where the num_features is 20 and the number of neighbors is 5. We choose such a setting because of the budget limit as it requires fewer perturbations and interactions with Instruct-GPT API. It generates more "sparse" saliency maps, we discuss this in detail in Section 6.2. The hyperparameters for GPT-2 and GPT-3 and

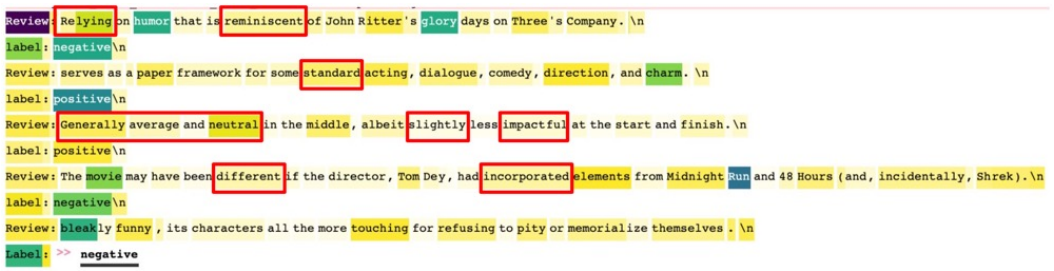
[1] distilbert-base-uncased-finetuned-sst-2-english
[2] https://huggingface.co/gpt2
[3] https://github.com/marcotcr/lime



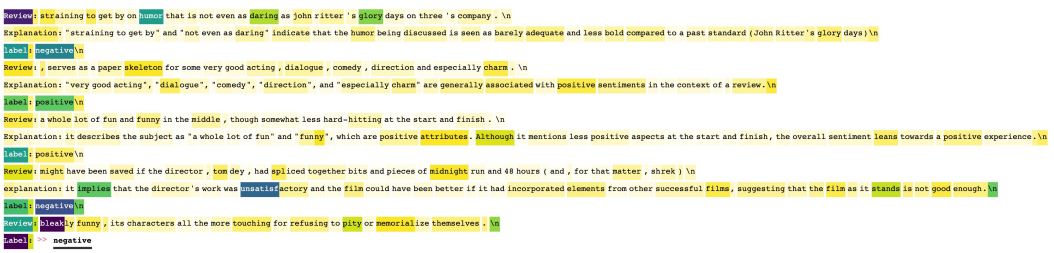
(a) Original prompt



(b) Prompt with label flipping in the demonstrations



(c) Prompt with input perturbation (neutralization) in the demonstrations



(d) Prompt with complementary explanations in the demonstrations

Fig. 2. Full prompts (demonstration + test example) used for original demonstration and three contrastive variants. Tokens are color-coded by saliency scores for GPT2 generated by IG. The red box in original and neutralized prompts indicates manually selected sentiment-indicative and sentiment-neutral terms that we used for saliency map comparison.

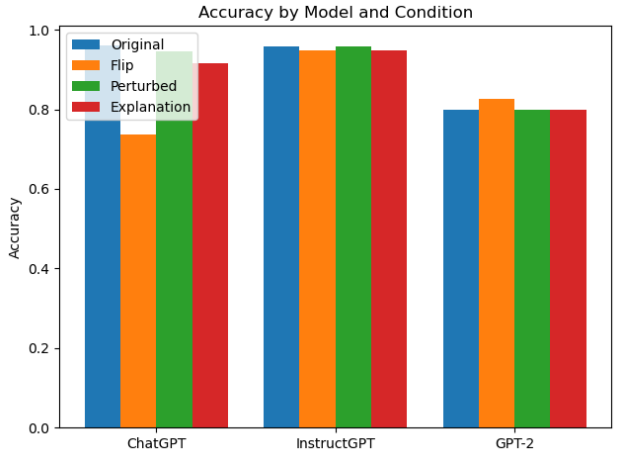


Fig. 3. Model Performance under the four conditions, with **four** demonstrations given

prompts are the same ones we used for the accuracy evaluation. We only generated saliency maps for GPT-2 and GPT-3 models due to time and compute constraints, future work could explore more models such as ChatGPT.

6 FINDINGS

6.1 Prediction Performance of the Three Contrastive Demonstrations

We tested the performance of GPT-3.5-Turbo, InstructGPT, and GPT-2 on the 288 selected test examples with four types of demonstrations, i.e., original, label flipping, input neutralization, and complementary explanations. The results are shown in Fig. 3 and Fig. 4 [4]. We find that for label-flipping demonstration, ChatGPT-Turbo-3.5 leads to the greatest degradation in performance, from accuracy 96% to 73% when given 4 demonstrations and to 17% when given 8 demonstrations. The performance of InstructGPT drops by a smaller amount in both scenarios. This is consistent with the findings from [Min et al. 2022; Wei et al. 2023], i.e., large LMs' (Instruct-GPT and ChatGPT) performance drops with increased number of the labels flipped in exemplars. Although the model parameter sizes are the same between GPT-3.5-Turbo and InstructGPT, GPT-3.5-Turbo seems to have a much stronger in-context-learning ability than InstructGPT.

We also noticed that GPT-2 model has much lower performance when given 4 demonstrations and almost predicts negative for all test examples when given 8 demonstrations. It is thus relatively insensitive to different types of contrastive demonstrations. The fact that the performance change of label-flipping is much smaller for GPT-2 compared with ChatGPT and Instruct-GPT also verifies the findings of emergent overriding ability of large LMS from [Wei et al. 2023].

On the other hand, we observe much smaller impacts for input neutralization and explanation. The small impact of input neutralization may be due to the fact that LMs are still able to make predictions using pretrained knowledge, especially for a relatively "easy" task of sentiment analysis. Adding complementary explanation to each examplar does not benefit the model performance for both four-demonstration and eight-demonstration scenarios. One of the reasons might be that the task chosen is too trivial for explanations to be useful.

[4]We excluded input perturbation from Fig. 4 because we perturbed the demonstrations differently while testing for 8 demonstrations (replace indicative words or phrases with opposite meaning words and phrases) and cannot afford to rerun the experiments in terms of time and cost.

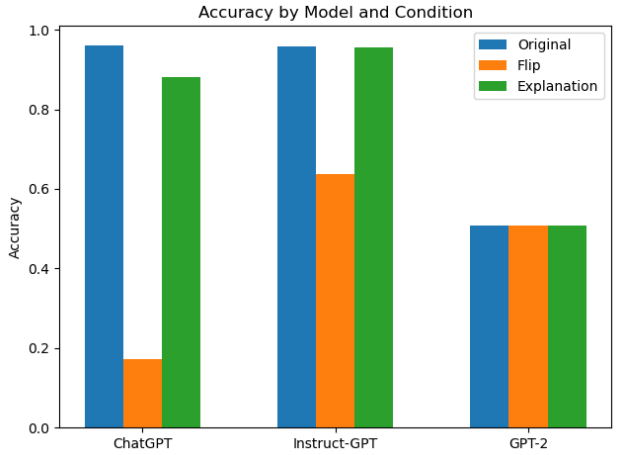


Fig. 4. Model Performance under the four conditions, with **eight** demonstrations given.

These mixed results further inspire us to contrast the patterns of the saliency maps generated by smaller LLMs and large LLMs as all the language models tested are built upon transformer architecture.

6.2 Comparison of the Saliency Maps

Due to the GPT-2's poor performance and compute cost when given 8 demonstrations, we use the setting of 4 demonstrations for saliency map comparisons as shown in Fig. 2 and Fig. 5.

6.2.1 Label Flipping.

Hypothesis. The labels in the demonstration are less important after model flipping for smaller LMs (GPT2) but more important for large LMs (`text-davinci-002 from Instruct-GPT`).

Analysis. For example as in Fig. 2a and Fig. 2b, the importance of the output label in the demonstration decreases from the original prompt to the label-flipped one. This suggests that the model might pay less attention to the flipped label due to its inconsistency with the input, which results in insensitivity to label flipping in the demonstrations. We expect smaller LMs (GPT2) and large LMs (`text-davinci-002` from Instruct-GPT) to have different behaviors because Wei et al. [2023] show only large LMs have the ability to override prior knowledge from pertaining to the one from demonstrations, which is also supported by our results from Fig. 3 and Fig. 4.

For GPT2, on average, 3.35/4 of the labels in the demonstration have decreased saliency scores when the demo labels are flipped. Moreover, the average saliency scores of the 4 demo labels **decrease** for all 20 test examples. The p-value from a T-test for comparing average saliency scores ($N = 20$) between original and label-flipped demonstrations is < 0.001. For InstructGPT, the average saliency scores **increase** for 16/20 test examples with a p-value of 0.23 from a similar T-test as above (Fig. 5b). As InstructGPT achieves around 60% accuracy in Fig. 4, we expect Instruct-GPT (with 8 demonstrations) and ChatGPT to have a more significant result as it shows the ability to fully override prior pretrained knowledge.

6.2.2 Input Perturbation (Neutralization).

Hypothesis. The sentiment-indicative terms in the original prompt are more important than sentiment-neutral terms in the neutralized prompt.

Analysis. The hypothesis is derived from the definition and our intuition of the sentiment analysis task. Sentiment-indicative terms are important to make sentiment predictions. To validate this hypothesis, we contrast the original and neutralized prompts and manually pick different tokens with sentiment orientations. The selected tokens are highlighted in Fig. 2a and Fig. 2c with red boxes respectively. We then compute the average saliency scores for each of the 20 test examples.

We find that, for GPT2, the average saliency scores for sentiment-indicative terms in the original prompt are higher than their contrastive parts in the neutralized prompt for all 20 test examples with a p-value of < 0.001 from a T-test. However, for Instruct-GPT, we find that the sentiment-indicative terms in the original prompt are equal or higher in 9/20 test examples with a p-value of 0.17 from a similar T-test as above. We note that, as mentioned in Section 5.3, the saliency maps for Instruct-GPT generated by LIME are sparse and have a lot of zeros as shown in Fig. 5. This may lead to a mixed result with a less significant T-test result.

6.2.3 Complementary Explanation. Previous work Ye et al. [2022] shows complementary explanations are beneficial for symbolic reasoning tasks including Letter Concatenation, Coin Flips, and Grade School Math. However, as we show in Fig. 3, complementary explanations do not necessarily improve the performance for sentiment analysis which is considered as an "easier" task for LMs. From saliency maps for GPT2, we find that 16/20 test examples have larger averaged saliency scores for tokens in the explanations than the ones in the review. On average, the averaged saliency scores for review tokens are 90% of the ones for explanation tokens. So the gaps are small and explanations are just as important as the original review in this sense. We suspect the impact of complementary explanations varies based on the tasks at hand. It may benefit tasks requiring more logical reasoning but such conclusion requires more systematical evaluation on more benchmark datasets and we leave it to future work.

7 LIMITATIONS

One potential limitation of our work is that we only select 288 samples, only 20 examples for the saliency map and only 5 neighbors in the LIME model. This is because of budge limitation and Openai daily request limit. In the future, a larger dataset is needed for the evaluation. Additionally, all the demos we used for the ICL is randomly selected, which will influence the model's accuracy performance in some cases. Therefore, a better method would be using different methods to select the demos, like similar instance retrieval. Finally, we only pick the saliency map as the explanation method. In the future, more explanation models will support our conclusion better. In addition, since we find out that adding explanations to the demonstrations leads to performance degradation, we would like to examine the role of explanation on other tasks, or with more examples being tested for the SST-2 dataset. However, due to time-constrains and budget issues, we are unable to conduct the experiment on more complex tasks, such as grade school math problems, or common sense reasoning to further examine why adding explanation leads to performance degradation on sentiment analysis task.

8 CONCLUSION

In this study, we used XNLP techniques to study how ICL works by investigating the performance of contrastive input-label demonstration pairs built in different ways, i.e., flipping the labels, changing the input, and adding complementary explanations, and contrasting their saliency maps with the original demonstration via qualitative and quantitative analysis. We partly reproduced the results from previous work [Min et al. 2022; Wei et al. 2023] on a sentiment classification task and found that the ground-truth labels in the demonstration are less salient after label flipping. We also found that neutralizing the sentiment-indicative terms in the input does not have as large impact as

changing ground-truth label do. The models may rely on pretrained knowledge to make fairly good predictions because the averaged importance scores for neutralized terms are smaller than the ones of original sentiment-indicative terms. Additionally, complementary explanations do not necessarily benefit sentiment analysis task as they do for symbolic reasoning tasks as shown in [Ye et al. 2022]. We hope the findings of this study can help researchers better understand the mechanism of LLMs and provide insights for practitioners when curating the demonstrations.

Future works can experiment with more LMs and more benchmark tasks and datasets to verify the findings of this study and make them more generalizable. Moreover, this work only focused on analyzing the saliency maps of demonstrations. Future work can investigate how demonstrations interact with the query test examples by adjusting exemplars in the demonstration based on their semantic distance from the query examples. It is also interesting to compare gradient-based saliency map methods with perturbation-based ones and see how the saliency maps differ with the same input and model.

REFERENCES

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2022. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661* (2022).

J Alammar. 2021. Ecco: An open source library for the explainability of transformer language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*. 249–257.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311* (2022).

Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. 2023a. Aligning Large Multi-Modal Model with Robust Instruction Tuning. *arXiv preprint arXiv:2306.14565* (2023).

Fuxiao Liu, Hao Tan, and Chris Tensmeyer. 2023b. DocumentCLIP: Linking Figures and Main Body Text in Reflowed Documents. *arXiv preprint arXiv:2306.06306* (2023).

Fuxiao Liu, Yinghan Wang, Tianlu Wang, and Vicente Ordonez. 2020. Visual news: Benchmark and challenges in news image captioning. *arXiv preprint arXiv:2010.03743* (2020).

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? *arXiv preprint arXiv:2202.12837* (2022).

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

K Simonyan, A Vedaldi, and A Zisserman. 2014. Deep inside convolutional networks: visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations (ICLR)*. ICLR.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825* (2017).

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, USA, 1631–1642. https://aclanthology.org/D13-1170

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 3319–3328. https://proceedings.mlr.press/v70/sundararajan17a.html

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846* (2023).

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080* (2021).

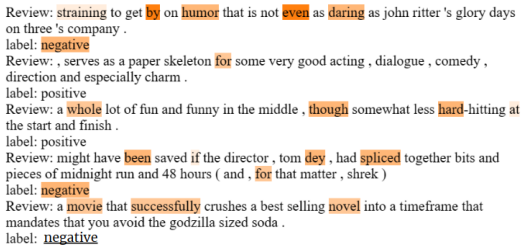
Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Ves Stoyanov, Greg Durrett, and Ramakanth Pasunuru. 2022. Complementary Explanations for Effective In-Context Learning. *arXiv preprint arXiv:2211.13892* (2022).

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2023. A Survey on Multimodal Large Language Models. *arXiv preprint arXiv:2306.13549* (2023).

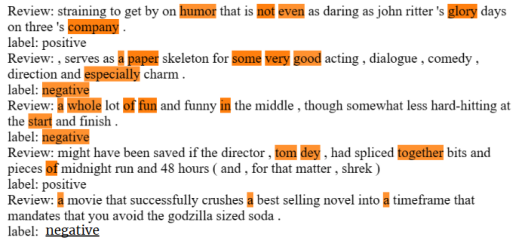
Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*. PMLR, 12697–12706.

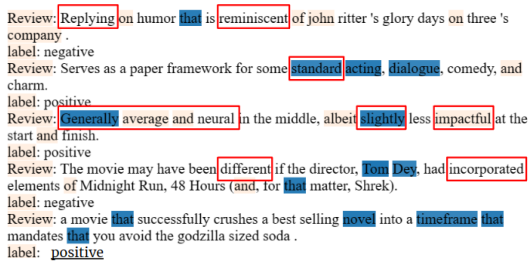
A EXAMPLE SALIENCY MAPS FOR INSTRUCT-GPT



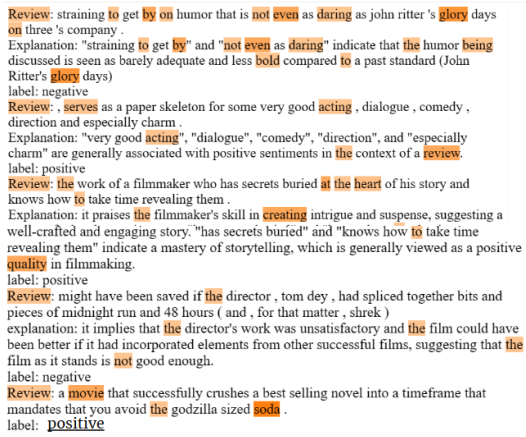
(a) Original prompts (demonstration + test example) used for original demonstration (Instruct-GPT)



(b) Prompt with label flipping in the demonstration (Instruct-GPT)



(c) Prompt with input perturbation (neutralization) (Instruct-GPT)



(d) Prompt with complementary explanations in the demonstrations (Instruct-GPT)

Fig. 5. Full prompts (demonstration + test example) used for original demonstration and three contrastive variants. Tokens are color-coded by saliency scores for generated by LIME. The red box in original and neutralized prompts indicates manually selected sentiment-indicative and sentiment-neutral terms that we used for saliency map comparison.