ML-708, Project, Final Report
Generic Prompts for Robust Vision-Language Models

Dmitry Demidov
MBZUAI
UAE, Abu Dhabi
dmitry.demidov@mbzuai.ac.ae

Mohammed El-Amine Azz
MBZUAI
UAE, Abu Dhabi
mohammed.azz@mbzuai.ac.ae

Raza Imam
MBZUAI
UAE, Abu Dhabi
raza.imam@mbzuai.ac.ae

Abstract

Vision language models have shown promising results in the fields of image classification and zero-shot learning. One of the most pre-trained models in this field is CLIP which relies on the introduced prompt during inference. This raises the issue of prompt optimisation and generalisation. The most popular methods have explored fine-tuning subsets of lists of prompt templates, while others have explored modifying the prompts with optimised words from dictionaries. Both approaches either suffer from much manual work and/or long training time besides unpredictable behaviour and limited generalisability. In our work, we explore the performance of CLIP with automated class-specific prompts instead of the manual fixed templates from previous works. We introduce a set of prompts that includes class labels to differentiate more between the classes during inference. The process requires a dataset of sentences that can be obtained via different means such as public datasets or text mining. Unlike previous work, our method is extremely fast and requires no training while achieving competitive results when compared to state-of-the-art methods. We show that our approach is generalisable to a larger number of classes while maintaining stable and good-quality results. It demonstrates promising behaviour that can be combined with CLIP's optimisation methods to further improve performance. We also make the code publicly available.

1 Introduction

Recently, vision-language pre-training [1] has emerged as a promising alternative for visual representation learning. The main idea is to align images and corresponding texts in a single feature space using separate encoders for each modality. This is also achieved by pre-training procedure with a large-scale dataset, which allows a model to learn more diverse visual concepts which can be easily transferred to another downstream task by only changing prompts [2, 3, 4]. More specifically, for a given classification problem, one first generate new classification weights with the text encoder by using task-relevant prompts for each category and then compare the weights with image features produced by the vision encoder.

Taking into account that most of the current state-of-the-art vision-language models are trained and tested using properly selected problem-specific fixed prompts, we can observe that text inputs may significantly affect the final performance. Nevertheless, while these models achieve remarkable results, they still highly rely on manual, localised input preparation, which raises the issue of

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

generalisation and robustness. In addition, prompt engineering also requires some prior knowledge about the type of language encoder and both pre-training and downstream datasets. Moreover, even with all the information available, identifying the right prompt and final tuning is a non-trivial task, which often takes a significant amount of time and resources.

However, as recent studies show [3], even with extensive tuning, the resulting curated manual prompts are not always optimal or even appropriate. So how can we make vision-language models more robust and the prompt selection process easier? Our proposed approach is to utilise automatically generated inputs from meta-level descriptions and randomly sampled sentences aligned with the coupled images, instead of the "hard" prompts. The key assumption is that by incorporating the higher-level textual descriptions for the images the model is introduced with more representative information, which may potentially positively influence the model's classification performance and robustness.

Therefore, in this work, we explore the CLIP's properties and behaviour under various automated ways of prompt generation. Moreover, besides the standard downstream quantitative analysis, we also investigate if our solution indeed affects the final prompts' quality via appropriate explainability tools.

2 Literature Review

Vision-Language Models. The need for expensive and massive labelled datasets has been gradually reduced by the vision-language models' ability to learn from text-image pairs that are freely accessible online. By simply focusing on pairing images and texts together in a joint embedding space [5], such models as CLIP have shown promising potential for a variety of classification tasks [1]. In detail, in order to function as a zero-shot classifier, it has an image encoder and a text encoder pre-trained to estimate which image is matched to its corresponding sentence with the correct class name from the dataset, as demonstrated in Figure 1.

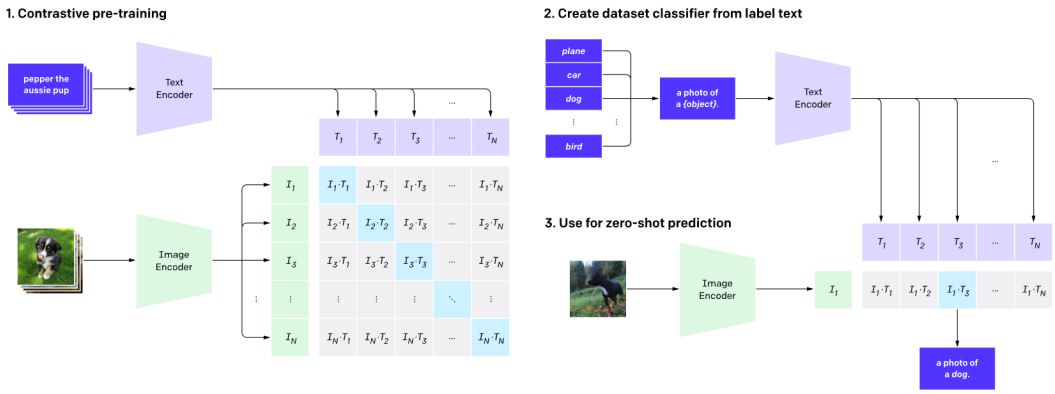

Figure 1: The overall architecture of CLIP-like models. Left: Training process. Right: Inference process. Source: [1].

As opposed to other SOTA ImageNet-trained models, CLIP can be configured to handle a wide range of visual classification tasks without the requirement for additional labelled training data. Some other derivatives of CLIP have also been presented in the recent literature: VirTex, ConVert, and ICMLM. The pre-training technique known as VirTex employs semantically rich captions to teach visual representations [5]. On COCO Captions, it is trained on convolutional networks from scratch before being used for later recognition tasks. ConVert is a model for automatical learning of medical visualizations from combinations of descriptive texts [6]. This technique uses a bidirectional objective between the two modalities to compare the visual representations with the related descriptive text. Image-conditioned masked language modelling (ICMLM) is a new proxy job on image-caption pairings that automatically generates image labels [7]. The visual representations learned as a byproduct of completing this task transfer effectively to a range of target tasks as ICMLM is made up of dedicated visual and textual encoders. For our tests, we will focus on a pre-trained Clip model and focus on the prompt engineering side of the experiments.

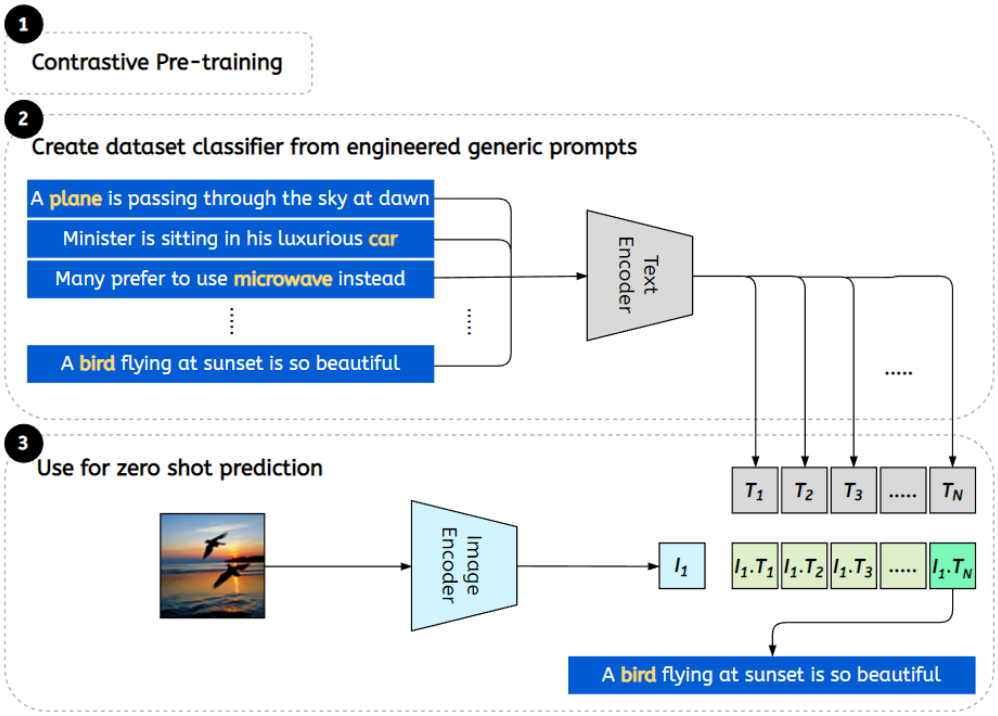

Figure 2: The inference process through CLIP on our engineered generic prompts. (1) implies contrastive pre-training which is same as the original CLIP. (2) Instead of using a set of templates for each class, we are passing the generated generic prompts to the text encoder to have respective text tokens. (3) Zero-shot prediction via cosine similarity between Image token (I_1) and respective Text tokens (T_i) are computed for prediction.

Prompt Learning. The topic of prompt learning comes from the NLP domain and represents the idea to construct text prompts that allow the Vision-Language model to understand the task more and be able to differentiate the classes better when inputting an image. In the [1] paper, the authors found that the simple usage of the sentence "A photo of {class}." is a good default prompt that improved their baseline slightly on ImageNet. However, the quality of these prompts can greatly affect the performance of the models, and generating good prompts manually can be time-consuming and difficult. To address this problem, the [2] paper proposes a method for automatic learning to generate better prompts for vision-language models. The method is based on the idea of using reinforcement learning to optimise the prompts based on the model's performance. The authors introduce a prompt generator network that takes as input an initial prompt and outputs a sequence of tokens that modify the prompt. As a follow-up work, [3] proposes a method that involves learning to generate effective prompts conditioned on the visual context, which helps the model generate more accurate and coherent outputs. The proposed method involves two stages: prompt generation and conditioning. In the generation stage, a prompt generator network is trained to generate a diverse set of prompts. In the conditioning stage, the prompts are conditioned on the visual features of the input image and the textual features of the prompt are fine-tuned to generate the desired output. While these methods perform well against previous and manual prompt engineering methods, they all require additional training and, therefore, are mostly not transferable or generalisable to other classification tasks.

3 Method

Hypothesis. Inspired by recent prompt learning research in NLP, in this work, our intention is to explore and study different ways of prompt preparation for vision-language models (such as CLIP). With this motivation, our proposed prompt engineering methodology aims to improve the robustness and generalization of the CLIP model by using meta-level descriptions instead of hard labels, and

by generating generic and descriptive prompts automatically. Our two main hypothesis for this approach is that: (1) The use of meta-level descriptions instead of hard labels in zero-shot learning through CLIP will improve transferability of the model to new datasets, and (2) Automatically gathered prompts to provide generic and descriptive prompts in inference through CLIP will improve robustness of the model. The goal is to make CLIP more adaptable to new and diverse datasets along with reducing the need for human intervention in prompt generation. This introduction of a more generic and descriptive label prompts may lead to a more general "understanding" of what is actually depicted in an input image.

Methodology. Taking the stated hypothesis into account, we argue that providing this kind of automatically gathered labels can be more straightforward and less human-intensive, since currently, most of the state-of-the-art prompt engineering techniques leverage either manually prepared sentences, where the "hard" label is put in, or generate a prompt by using learnable tokens. While most of the recent works related to vision-language models [2, 3, 4] have explored various ways of prompt generation, in our work we emphasise that such techniques, while being effective, may not add a considerable amount of meaningful and potentially generalisable information. Instead, we investigate the model's behaviour when more implicit information (such as a dictionary description or sentence use) is utilised as a label. Moreover, unlike other works which use prompts based on a general and unified context shared among all classes, we suggest using prompts based on detailed and class-specific context suitable for each class. Specific details about the prompts preparation can be found in Section 4.1.

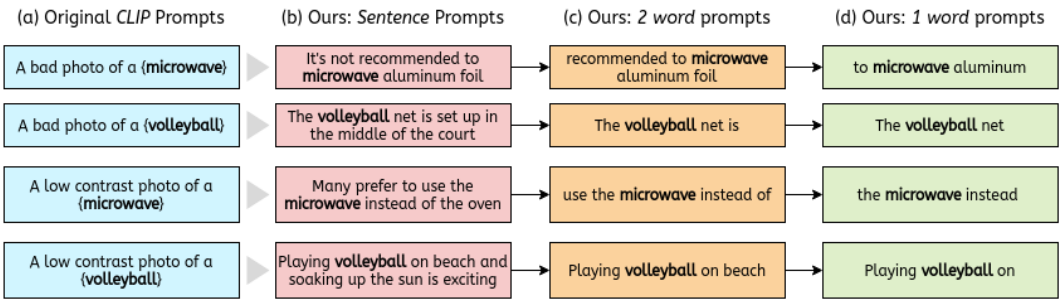

Figure 3: In our prompt generation, instead of using templates (a) for every class as in the original CLIP, we are using class-specific generic prompts (b) for each class. Further, we extract prompts with n-word to the left and right of the class word ((c) and (d)) from the originally generated full-sentence prompts.

Our rationale for the proposed hypothesis is that: (1) Using meta-level descriptions are more flexible and abstract, allowing the model to capture more general features of the images and text. Therefore, the model trained on this methodology will be more robust to variations in data distribution and able to handle diverse datasets with better performance than models trained on hard labels. (2) Moreover, automatically gathered prompts are based on a large and diverse corpus of text and can provide a more comprehensive understanding of the data than manually prepared templates, sentences or learnable tokens. Therefore, the model trained on this methodology will be more adaptable and transferable to new datasets and require less human intervention in the training process, resulting in a more flexible and robust model (Figure 4). Based on the stated hypothesis, our ultimate goal consists of two core parts: (1) investigation of the ability of our approach to provide prompts that lead to the model's robustness and performance improvement and (2) elimination of the need to manually prepare more "meaningful" prompts (with class-specific context). In case of any, negative or positive, results, we also would like to study the underlying reasons for the obtained outcomes through explainability-providing methods. Figure 2 illustrates the inference process done on our engineered generic prompts.

Generic Prompts. Our approach involves using the ImageNetV2-1k validation classes and the UMBC web-based corpus of paragraphs to train a pre-trained CLIP model with a ViT-B/32 image encoder. The goal of this prompt engineering approach is to improve the robustness and generalisation of CLIP by using meta-level descriptions instead of hard labels and providing generic and descriptive prompts to the model. The details of our approach to generating these generic prompts can be described by the following steps:

1. For each class in a dataset, search for k number of sentences which include the class name.

2. Out of the found sentences, prepare shorter prompts with n words around the class name and filter out ambiguous texts.

3. Use a CLIP's text encoder to generate feature vectors for each prompt and average them for each class.

4. Prepare zero-shot weight to create a standard CLIP classifier that maps both the images and texts into a shared feature space.

5. Performance of the CLIP model and quality of the generated prompts can be further evaluated along with the transferability and robustness of the solution.

6. This methodology can be further applied to any datasets and classes, as long as appropriate prompts are generated for each class.

The length of prompts affects the performance of CLIP inference as it is possible that longer prompts may be more susceptible to noise, as they may contain more irrelevant or misleading information. Shorter prompts, on the other hand, may be less affected by noise, as they provide less opportunity for irrelevant information to be included. This implies that the length of prompts may affect the interpretability of the model's predictions, as shorter prompts may be more easily understood by humans.

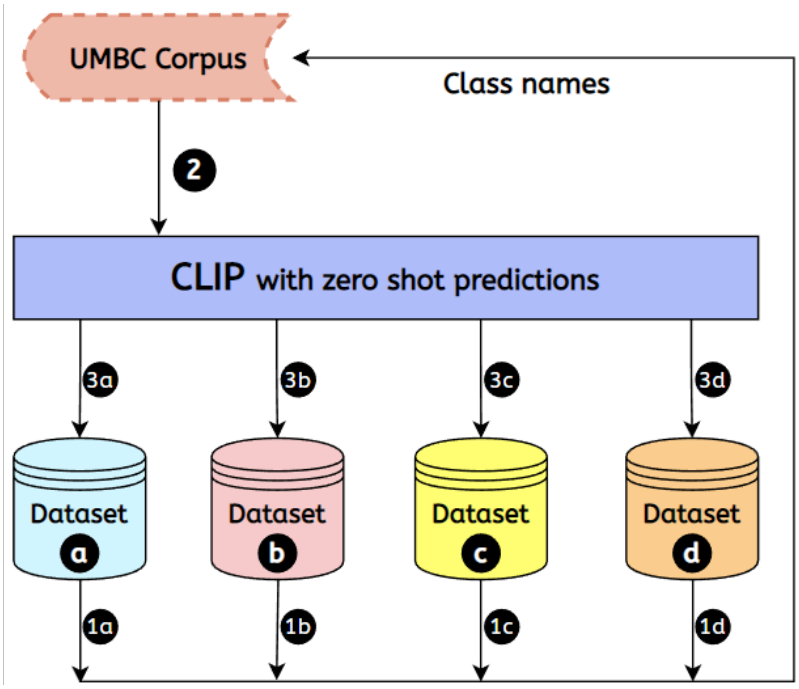

Figure 4: Expected transferability of our proposed approach.

Therefore, selecting the appropriate prompt length is crucial for ensuring that the model is able to generalize well to new data while minimizing the impact of noise and maximizing interpretability. In our proposed methodology, along with generating the full prompt sentences, we are also generating different prompt sizes using the original full prompts with the aim to explore the effect of the length of different prompts. We generate n-word prompts by taking the first n words to the left and right of the class word in the full prompt. This results in a new prompt with a total of $2n + 1$ words. Figure 3 explains our extraction of n-word out of the originally generated full prompt for a specific class.

4 Experiments and Results

4.1 Setup

In all our experiments, we utilise pre-trained CLIP with the ViT-B/32 image encoder, unless stated the opposite, and use the following datasets:

- ImageNetV2-1k [8] validation set (and its subsets of 12 and 100 random classes).
- UMBC web-base corpus [9] consisting of various paragraphs of English text. We extract corresponding sentences containing ImageNet class names.

In our approach, for each class, we prepare 100 or 1000 prompts for each class, while CLIP authors suggested using only 80 prompts per class for the ImageNetV2 dataset. We group the prompts by class and then process each prompt in each class, where CLIP further maps both images and texts into a shared feature space. The pre-processing of images involves only resizing to 224x224 and normalisation. We pass all text-image pairs to CLIP for all classes, resulting in $P_n * N_{cls}$ prompts, where P_n is the number of prompts per class and N_{cls} is the number of classes.

Through the experiments, we validate the performance of the proposed approach and its behavior under different classification regimes. In addition, we explore the influence of its hyper-parameters: the length of prompts, the number of prompts per class, and rules for filtering the prompts.

4.2 Quantitative Analysis

Procedure. For quantitative analysis, we study CLIP's classification performance in two different setups where both are conducted with the ImageNet validation set. For the first setup, we only use 100 classes and perform a 100-hot classification, while for the second one, we also use the same 100 classes but perform a 1000-hot classification so that we change classification weights for our picked 100 classes only and the weights for the rest of the classes are from original CLIP.

In both setups, we investigate the performance of CLIP with two types of prompts. The first one is our pure approach, where we only use k words around the class name in the found sentence from the text corpus, where k is from 1 to 5 words. The second type of prompt uses the same generated from the first type, but now it is put into a template *"a photo of {}"*, which is a general template recommended by the CLIP authors. The evaluation procedure is identical for both types of prompts and includes further averaging and zero-shot weights creation.

Results. As we can see in Table 1, in the more fair setting of the 100-classes-only classification, our approach with the best setup demonstrates better performance with a classification difference of about 2 %., while still being better even without borrowing the "A photo of {class}" template from the CLIP paper. Moreover, with the less fair 1000-hot classification, the solution with our prompts remains dominant showing impressive 6 % growth with the best setup.

4.3 Qualitative Analysis

For the qualitative analysis of our solution, we performed a comparative evaluation of the similarity scores between our and CLIP's prompts.

Similarity Score. The similarity score is a metric used to measure the similarity between two images or an image and a text prompt. The similarity score for CLIP is calculated by feeding the two images or an image and a text prompt into the model, which then computes a vector representation of each input. The similarity score is then obtained by measuring the cosine similarity between the two vectors, which ranges from -1 (completely dissimilar) to 1 (identical), and then rescaled keeping the positive values and crushing the negatives to 0. The higher the similarity score, the more similar the two inputs are considered to be by the CLIP model. As such, The similarity score can be used to check which prompts are closer to the inputted image, which can then be used to classify it. The Equation for calculating the Similarity score is:

$$Score(I, C) = max(100 * cos(E_I, E_C), 0), \tag{1}$$

where E_I is the CLIP embedding for an image I and E_C the embedding for the text C [10].

Table 1: Quantitative results with CLIP model using original ImageNet-specific and our generated prompts in different classification regimes. The numbers marked with "$*$" are obtained using our averaged prompts for 100 initially picked classes, while the prompts for the rest 988 classes are original ImageNet-specific ones. The results for the CLIP with cherry-picked manual prompts are for reference. Results of our method outperforming CLIP with the base prompt are highlighted in bold, while our best results are also underlined.

Method	**ImageNet Validation Set**			
	100 classes among 100		**100 classes among 1000**	
	Top-1	***Top-5***	***Top-1***	***Top-5***
CLIP (Manual Prompts)	*84.9*	*97.2*	*59.3*	*85.6*
CLIP ("A photo of {class}")	80.8	**96.1**	55.6	81.3
Ours (100 prompts, 1 word)	79.7	94.6	44.3*	76.4*
Ours (100 prompts, 2 words)	79.1	95.4	43.6*	74.4*
Ours (1000 prompts, 1 word)	**80.9**	95.4	44.3*	77.7*
Ours (1000 prompts, 2 words)	79.7	95.6	43.9*	76.8*
Ours (100 prompts, 1 word) + "A photo of {prompt}"	**81.1**	95.3	**59.3***	**84.0***
Ours (100 prompts, 2 words) + "A photo of {prompt}"	79.8	95.2	**60.2***	**85.1***
Ours (1000 prompts, 1 word) + "A photo of {prompt}"	**82.0**	95.8	**60.3***	**85.8***
Ours (1000 prompts, 2 words) + "A photo of {prompt}"	**82.1**	95.9	**61.7***	**86.0***

Procedure. For the qualitative evaluation between our and CLIP's prompts, we randomly sampled 12 classes out of 1000, namely "cucumber," "mushroom," "banana," "pizza," "bucket," "umbrella," "mailbox," "microwave oven," "rifle," "torch," "volleyball," and "taxicab." More specifically, from the validation set we load one random image for each class and prepare prompts for each of the 12 classes, the same way as described in Section 3.

In the experiments, we compute the cosine similarity scores between each prompt and its corresponding image. We group the prompts by class and then calculate similarity scores for each prompt using cosine similarity. To ensure fairness in comparative assessment, we evaluated the minimum, maximum, mean, and median cosine similarity scores for each class for both 100 and 1000 prompts. The results are presented in Figure 5, and additional results can also be found in Appendix A.1.

Results. The experiment compared the similarity scores between our prompts and the manually curated prompts for the ImageNet classes. We can observe that the difference is fluctuating among the classes. For example, our approach showed better performance than the CLIP one for the class "cucumber" with higher, indicating that our curated prompts are more similar to the cucumber class image. On the other hand, the mean cosine similarity score for the "microwave oven" class was higher for the CLIP manual prompts, indicating that the ImageNetV2 dataset contains images of microwave ovens that can be better described by the CLIP prompts.

The results also show that the similarity scores varied significantly for different classes. For example, the median similarity score for the "mushroom" class was 26.30, while the median score for the "taxi" class was only 22.56. This indicates that some classes may be more challenging to curate than others and that the performance of the manual prompts may vary depending on the classes included. Meanwhile, our automated prompts provide more stable scores.

In addition, we also noticed that our generated prompts with 1/2/3/5 words around the class name in the found sentence provide various similarity scores for the same images, which means that prompt length also significantly affects overall performance. The general trend shows that the mean cosine

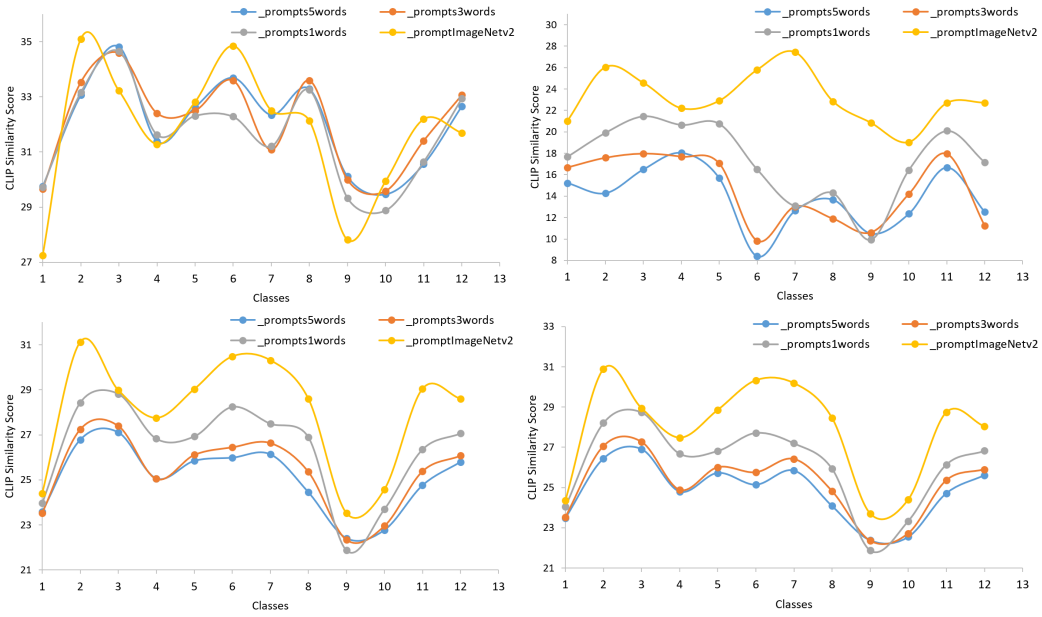

Figure 5: Results assessing the similarity scores across 1000 prompts for each class. Plots demonstrate averaged similarity scores for all prompts variants across all 12 classes, where Top Left (a) shows the maximum scores, Top Right (b) shows the minimum scores, Bottom Left (c) shows the median of scores, and Bottom Right (d) shows the average of the scores.

similarity score is higher when using 1-word or 2-word prompts compared to 3- and 5-word ones. However, more research is required to explore the effect of prompt length on the similarity quality.

5 Conclusion and Future Work

In this work, we have explored the use of meta-level class-specific prompts to improve the accuracy of CLIP during inference. Our proposed solution achieves stable and good-quality results outperforming the manual CLIP prompts in most cases. Through extensive experiments, we found out that the optimal prompt's length is 3-10 words and the optimal number of prompts per class is between 100-1000 prompts. The main and most important benefit of our solution is that, unlike other methods, it does not require any training process or active and long manual work, since it automatically prepares a dictionary of prompts.

We also emphasize that our solution can be further improved by the following potential optimisations:

- The results can be further improved by tweaking and polishing the search pipeline;
- The solution can work incredibly faster if optimised for multi-core processing;
- A better class names pre-processing can be done (e.g., search by more common synonyms to find sentences easier);
- The final prompts can be potentially improved by filtering them out in a better way (e.g., remove unnecessary words such as "and", "or", etc.);
- Picking only Top-N prompts out of all the generated prompts (e.g., top-10 %);

For this purpose, we also release the code of our solution to the public.

Regarding the future work, we consider the following directions:

- Adaptation of our solution to all 1000 ImageNet classes;
- Applying our approach to other datasets (e.g., CIFAR, Fine-Grained datasets, etc.);

A Appendix

A.1 Additional Results

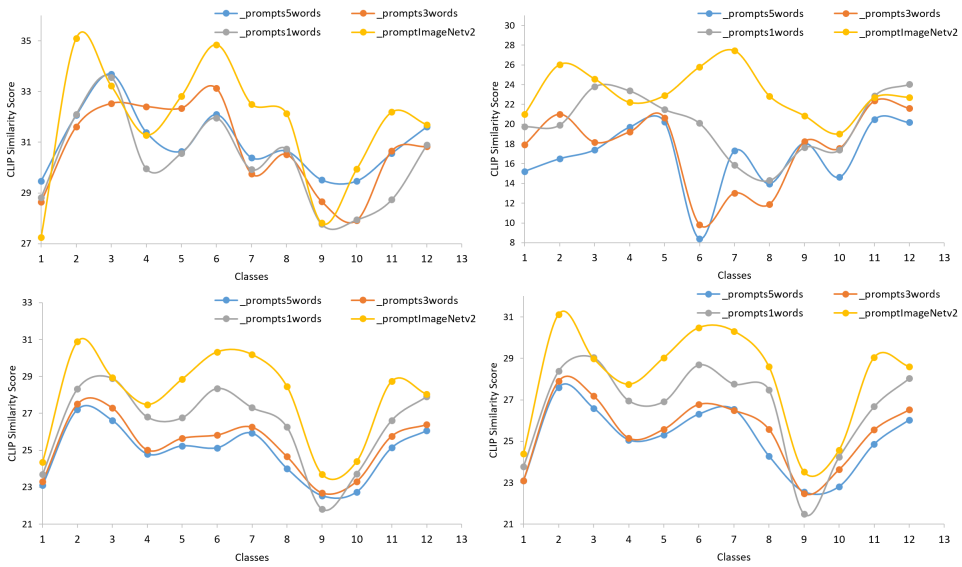

Figure 6: Qualitative results assessing the similarity scores across 100 prompts for each class. Plots demonstrate averaged similarity scores for all prompts variants across all 12 classes, where Top Left (a) shows the Maximum scores, Top Right (b) shows the Minimum scores, Bottom Left (c) shows the average of scores, and Bottom Right (d) shows the median of the scores.

References

[1] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[2] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, jul 2022.

[3] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models, 2022.

[4] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning, 2022.

[5] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11162–11173, 2021.

[6] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747*, 2020.

[7] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning visual representations with caption annotations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 153–170. Springer, 2020.

[8] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *CoRR*, abs/1902.10811, 2019.

[9] Tim Finin James Mayfield Lushan Han, Abhay L. Kashyap and Johnathan Weese. UMBC EBIQUITY CORE: Semantic Textual Similarity Systems. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics, June 2013.

[10] Nicki Skafte Detlefsen, Jiri Borovec, Justus Schock, Ananya Harsh, Teddy Koker, Luca Di Liello, Daniel Stancl, Changsheng Quan, Maxim Grechkin, and William Falcon. TorchMetrics - Measuring Reproducibility in PyTorch, 2 2022.