Training-Free Semantic Segmentation via LLM-Supervision

Wenfang Sun[1] Yingjun Du[2] Gaowen Liu[3] Ramana Rao Kompella[3] Cees G. M. Snoek[2]

[1]University of Science and Technology of China [2]University of Amsterdam [3]Cisco

Abstract

This paper introduces a new approach to text-supervised semantic segmentation using supervision by a large language model (LLM) that does not require additional training. Our method starts from an LLM, like GPT-3, to generate a detailed set of subclasses for a more accurate class representation. We then employ an advanced text-supervised semantic segmentation model to apply the generated subclasses as target labels, resulting in diverse segmentation results tailored to each subclass's unique characteristics. Additionally, we propose an assembly that merges the segmentation maps of the various subclass descriptors to ensure a more comprehensive representation of the different aspects in the test images. Through comprehensive experiments on three standard benchmarks, our method outperforms traditional text-supervised semantic segmentation methods.

1. Introduction

Semantic segmentation plays a crucial role in computer vision, aiming to assign specific semantic classes to their respective pixels. The recent emergence of vision and language models [7, 8, 12], as a standard for generalized zero-shot learning, has given rise to a new area of research: language-driven semantic segmentation [6, 15], which enables the creation of semantic segments through contrasting image and text. Recently, some studies [14, 17] have investigated segmentation based on CLIP to enhance transferability, focusing on areas such as zero-shot [2, 13] and open-vocabulary segmentation [16]. These descriptors are key to the accuracy and effectiveness of language-driven semantic segmentation models, and the focus of this paper.

We are inspired by recent work on training-free vision language models, *e.g.*, [9, 11]. These models utilize large language models (LLM) [1] to generate additional descriptions per class, such as subclass names [9], class descriptions [11], or concept descriptions [10]. These approaches are based on the generation of additional descriptions to provide each class with more detailed and informative representations. However, their application in semantic segmentation tasks has not yet been investigated, which is more challenging due to the precise pixel-level understanding and differentiation of various image elements. In this paper, we propose a novel training-free semantic segmentation via LLM-supervision.

We make three contributions. First, we suggest employing LLMs to generate subclass names for each class to tackle the problem of similar semantic features in categories found in traditional text-guided semantic segmentation techniques. Second, we implement an advanced text-supervised semantic segmentation method [15], using the generated subclass descriptors as target labels. This approach facilitates the production of a wide variety of segmentation results. Third, we introduce the ensembling of subclass descriptors, efficiently combining segmentation maps from different subclass descriptors with the original superclass representation. We demonstrate that our method outperforms conventional text-supervised segmentation methods through extensive evaluation across three commonly used standard benchmarks.

2. Methods

2.1. Generation of the subclass with an LLM

We use the GPT-3 model [1] to generate detailed subclasses, which provide more informative and distinguishable features for each class. This approach also reduces the dependence on human experts for subclass generation, aligning with our objective of fully automating this process, inspired by the concepts in [9]. GPT-3 demonstrates substantial knowledge in the area of subclasses, showing effectiveness in representing each class when given appropriate prompts. Specifically, for a set size n of subclasses and a given superclass name `class-name`, we prompt GPT-3 with $\mathcal{P}_1$:

$\mathcal{P}_1$: General subclass generation guide

```
Q: List n subclasses of the
following: class-name
A: Here are n commonly seen
subclasses of class-name:
```

Here n is the number of the generated subclass name, we set $n = 10$ in this paper.

To optimize performance with the above prompt, we supply GPT-3 with two examples of desired outputs for few-shot

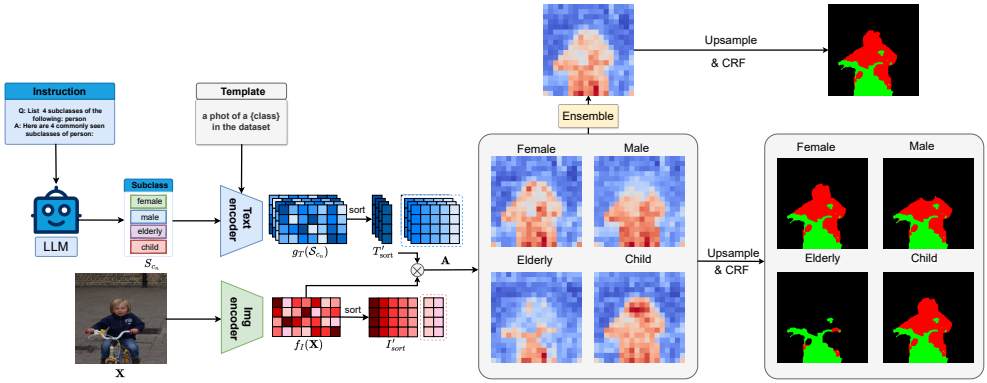

Figure 1. Overall framework of our training-free semantic segmentation via LLM supervision.

adaptation. Notably, these examples can be standardized and applied consistently across all datasets, eliminating the need for additional user input when generating a subclass set for a new dataset. We show the optimized prompt $\mathcal{P}_2$ as follows:

$\mathcal{P}_2$: Subclass generation guide with few examples

```
Q1:List 3 subclasses of the person:
A1:female, male, child
Q:List n subclasses of the following:
class-name
A:Here are n commonly seen subclasses
of class-name:
```

After acquiring the subclasses generated by GPT-3, our goal is to use these subclasses to supervise each image for more accurate predictions. We then discuss how to apply these subclasses in an advanced text-supervised semantic segmentation approach in the next section.

2.2. Training-free semantic segmentation

We leverage SimSeg [15] as our baseline, which is the state-of-the-art data-free text-supervised semantic segmentation. Here, each superclass label, denoted as c, represents a specific concept in natural language, *e.g.*, *person*. We then input each superclass label c into the GPT-3 to generate a list of subclass set $\mathcal{S}_{c_n}$, *e.g.*, $\{female, male, elderly, child\}$, where c_n is the $n-th$ subclass name of superclass c. We proceed by entering all the subclasses' names generated into the text encoder using specific predefined templates, such as *a photo of a {subclass}*. We employ two encoders from SimSeg [15], namely f_I for processing image data I and g_T for processing text data T. So the features of the generated subclass are represented as $[g_T(\mathcal{S}_{c_n})] \in \mathbb{R}^{n\times m_t\times d}$. Next, we input the test image X into the image encoder to extract image features, denoted as $f_I(\mathbf{X}) \in \mathbb{R}^{m_i\times d}$. In line with SimSeg [15], we also adopt the locality-driven alignment (LoDA) technique using the feature selection method of maximum response selection. This method helps avoid the overly dense alignment of pixels and entities during the optimization of CLIP. The textual features are then arranged in descending order along the dimension d: $T_{\text{sort}} \in \mathbb{R}^{n\times m_t\times d} = \text{sort}_d([g_T(\mathcal{S}_{c_n})])$. By doing so, LoDA selects the features that exhibit the highest values in each channel as $T'_{\text{sort}} \in \mathbb{R}^{n\times d}$, anticipated to encompass vital local textual concepts. Subsequently, we determine the resemblance between image features $f_I(\mathbf{X})$, and sorted text features T'_{sort}, to derive the attention weight $\mathbf{A}$. This weight will play a crucial role in the ensemble phase. The calculation of the attention weight $\mathbf{A}$: $\mathbf{A} \in \mathbb{R}^{n\times m_i} = [f_I(\mathbf{X})] \times T'_{\text{sort}}$. Subsequently, we generate an n initial coarse mask by applying attention weight $\mathbf{A}$ to each pixel of the original image.

To achieve more accurate mask results, we implement several post-processing steps that do not require additional training. Initially, we employ upsampling techniques to meticulously restore complex details in the image, which helps in recovering information that might have been lost, thereby improving the visual accuracy of the upsampled features. Subsequently, we utilize the conditional random field (CRF) [5] method as a crucial step to refine the outcomes of the upsampling process. The inclusion of CRF aids in solidifying the enhanced features from upsampling, enabling the effective capture and use of expanded pixel relationships. This step is pivotal in thoroughly understanding the complex structures present in the image.

Finally, we achieve finely segmented results for n distinct subclasses within the tested superclass. Following this, we delve into the fusion of textual data across various subclasses. The purpose of this fusion is to augment the overall semantic interpretation of the image, leading to segmentation masks of higher precision. The overall framework is shown in the Figure 1. The middle four images are the coarse masks before upsampling and CRF.

2.3. Ensembling of subclasses descriptors

We propose subclasses descriptor ensembling to merge segmentation results from various subclasses, thereby achieving more precise segmentation results. We initially perform feature selection on image features $[f_I(\mathbf{X})] \in \mathbb{R}^{m_i\times d}$:

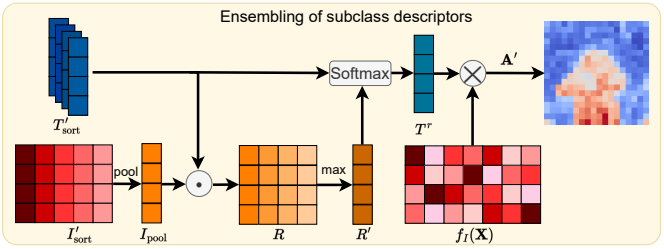

Figure 2. Ensembling of subclasses descriptors.

$I_{\text{sort}} \in \mathbb{R}^{m_i \times d} = \text{sort}_d([f_I(\mathbf{X})])$. Differing from the selection of textual features T'_{sort}, we choose locally responsive features $I'_{\text{sort}} \in \mathbb{R}^{5 \times d}$ from the top 5 dimensions with maximum responses, and discard the left features. This enables us to encompass important visual concepts and key entities within the image. We apply an average pooling operation to the image features I'_{sort}. Following this, we perform element-wise multiplication (Hadamard product) on the pooled image features $I_{\text{pool}} \in \mathbb{R}^d$ and the textual features T'_{sort}, yielding relationship weights $R \in \mathbb{R}^{n \times d}$ between them: $R \in \mathbb{R}^{n \times d} = I_{\text{pool}} \cdot T'_{\text{sort}}$ Next, we calculate the row-wise maximum in the relationship features R to obtain $R' \in \mathbb{R}^n$, which fuses important information from descriptors of different subclasses. Then, we apply the softmax operation to R' and the textual features T'_{sort}. Consequently, we obtain the final fused textual features $T^r \in \mathbb{R}^d$, which are generated based on the alignment between the image features and the textual features from different subclasses. Finally, we calculate the similarity between the textual features T^r, which contain rich subclass semantic information, and the image features $[f_I(\mathbf{X})]$, to obtain the final attention weights $\mathbf{A}'$: $\mathbf{A}' \in \mathbb{R}^{m_i} = [f_I(\mathbf{X})] \times T^r$. Using the $\mathbf{A}'$, we then perform upsampling and CRF post-processing operations, ultimately generating a more precise mask. Our ensemble process is simple and efficient, requiring no parameter updates, thereby achieving a training-free semantic segmentation approach. We show the framework of ensembling of subclasses descriptors in Figure 2.

3. Experiments

Benefit of LLM-supervision. Table 1 presents a comparison between our model and various baselines, showcasing its effectiveness. In this experiment, we apply our LLM-supervision technique to the advanced text-supervised semantic segmentation method, SimSeg [15], which lacks image-mask pairs during training. By integrating our LLM-supervision, there is a consistent and substantial improvement in performance across different datasets. Notably, on the PASCAL VOC dataset, our approach surpasses SimSeg by a significant 5.1% margin. Moreover, the results using $\mathcal{P}_2$ are consistently better than those with $\mathcal{P}_1$. This indicates that using a few-shot sample approach can improve the generalization of the created subclasses, thereby enhancing overall performance. Additionally, we have included visualizations of the segmentation outcomes in Figure 3. The segmentation results using only superclass text-supervision tend to concentrate on the general characteristics of each class. In contrast, our LLM-supervision approach is capable of capturing more detailed information, thanks to the generated subclasses. For instance, while SimSeg, using the superclass *cat*, can only segment the head of the *cat*, our model successfully segments both the head and body of the *cat*. These improvements are attributed to the LLM-supervision, which generates more informative class representations, resulting in consistent enhancements over the use of superclass textual representations alone.

Subclass quality impact. In our study, we used subclass text as supervision to enhance our model's performance in segmentation. The effectiveness of this approach is evident in the per superclass results on the Pascal VOC dataset, as shown in Table 2. Our model, with LLM-supervision, outperforms SimSeg [15] significantly in average results. This improvement varies across classes, depending on the quality of the generated subclasses. A notable example is the *person* superclass, where our model achieves a 50.4% score. This is attributed to the distinct features of subclasses like *male*, *female*, and *child*. However, for classes like *sofa*, our model underperforms due to less distinct subclasses. Future efforts will focus on refining subclass quality to further enhance our model's efficiency and applicability. Our findings confirm that the quality of generated subclasses is critical for effective text-supervised semantic segmentation.

4. Conclusions

We introduce a novel text-supervised semantic segmentation approach leveraging large language model supervision, eliminating the need for additional training. This method begins with the use of an LLM to generate a rich set of subclasses for enhanced class representation. We incorporate these subclasses as target labels into a sophisticated text-supervised semantic segmentation model, leading to a range of segmentation results that mirror the distinct characteristics of each subclass. Further, we propose ensembling subclass descriptors to merge segmentation maps derived from different subclass descriptors, ensuring comprehensive coverage of diverse aspects within test images.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020. 1

[2] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. *NeurIPS*, 32, 2019. 1

Backbones	Models	Pre-training Dataset	Supervision	Zero-shot	Transfer PASCAL VOC	Transfer PASCAL Context	Transfer COCO-Stuff
ViT-S	DINO [3]	ImageNet	self	✗	39.1	20.4	-
ViT-S	MoCo [4]	CC3&12M+YFCC	self	✗	36.1	23.0	-
ViT-S	SimSeg [15]	CC3&12M	text	✓	56.6	25.8	27.2
ViT-S	**Ours (with $\mathcal{P}_1$)**	CC3&12M	text	✓	60.6	27.2	28.5
ViT-S	**Ours (wtih $\mathcal{P}_2$)**	CC3&12M	text	✓	**61.7**	**27.8**	**29.1**

Table 1. Benefit of LLM-supervison for text-supervised semantic segmentation.

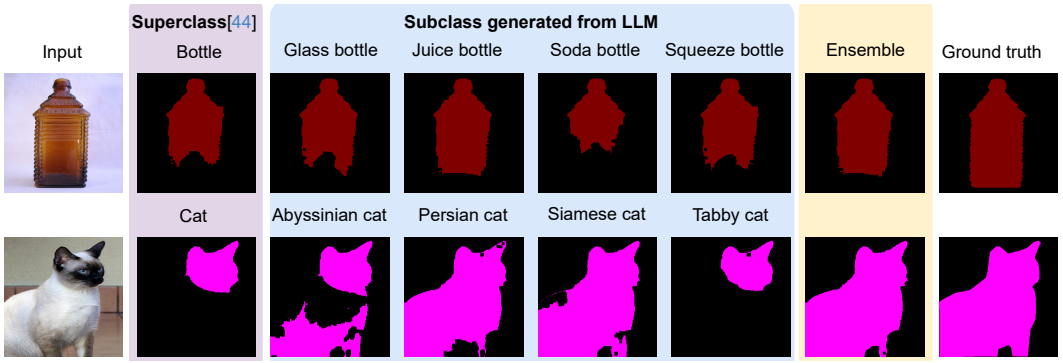

Figure 3. Segmentation Results with Our LLM-Supervision.

Models	person	boat	aeroplane	cow	monitor	table	bottle	horse	car	bus	train	cat	dog	bicycle	plant	bird	chair	sheep	motorbike	sofa	Avg
SimSeg	39.0	45.4	72.6	67.3	36.2	26.2	41.6	69.2	62.8	74.8	56.9	79.4	74.3	34.0	39.7	80.3	16.8	74.9	70.9	**42.4**	56.6
Ours	**50.4**	**55.4**	**82.6**	**76.7**	**44.4**	**33.9**	**48.2**	**75.4**	**68.1**	**79.3**	**61.3**	**83.6**	**78.4**	**37.2**	**42.4**	**82.9**	**19.2**	**77.1**	**71.1**	38.6	**61.7**
Δ	+11.4	+10.0	+10.0	+9.4	+8.2	+7.7	+6.6	+6.2	+5.3	+4.5	+4.4	+4.2	+4.1	+3.2	+2.7	+2.6	+2.4	+2.2	+0.2	-3.8	+5.1

Table 2. Class-wise IoUs on Pascal VOC.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *CVPR*, pages 9650–9660, 2021. 4

[4] X Chen, S Xie, and K He. An empirical study of training self-supervised vision transformers. In *ICCV*, pages 9620–9629, 2021. 4

[5] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *NeurIPS*, 24, 2011. 2

[6] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022. 1

[7] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, pages 12888–12900. PMLR, 2022. 1

[8] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *ICLR*, 2022. 1

[9] Zachary Novack, Julian McAuley, Zachary Chase Lipton, and Saurabh Garg. Chils: Zero-shot image classification with hierarchical label sets. In *ICML*, pages 26342–26362. PMLR, 2023. 1

[10] Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *ICLR*, 2023. 1

[11] Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *ICCV*, pages 15691–15701, 2023. 1

[12] 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 *ICML*, 2021. 1

[13] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero-and few-label semantic segmentation. In *CVPR*, pages 8256–8265, 2019. 1

[14] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *ECCV*, pages 736–753. Springer, 2022. 1

[15] Muyang Yi, Quan Cui, Hao Wu, Cheng Yang, Osamu Yoshie, and Hongtao Lu. A simple framework for text-supervised semantic segmentation. In *CVPR*, pages 7071–7080, 2023. 1, 2, 3, 4

[16] Hang Zhao, Xavier Puig, Bolei Zhou, Sanja Fidler, and Antonio Torralba. Open vocabulary scene parsing. In *ICCV*, pages 2002–2010, 2017. 1

[17] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *ECCV*, pages 696–712. Springer, 2022. 1