LLM4SGG: Large Language Models for Weakly Supervised Scene Graph Generation

Kibum Kim[1] Kanghoon Yoon[1] Jaehyeong Jeon[1] Yeonjun In[1] Jinyoung Moon[2]
Donghyun Kim[3] Chanyoung Park[1*]
[1]KAIST [2]ETRI [3]Korea University
{kb.kim,ykhoon08,wogud405,yeonjun.in,cy.park}@kaist.ac.kr jymoon@etri.re.kr, d_kim@korea.ac.kr

Abstract

*Weakly-Supervised Scene Graph Generation (WSSGG) research has recently emerged as an alternative to the fully-supervised approach that heavily relies on costly annotations. In this regard, studies on WSSGG have utilized image captions to obtain unlocalized triplets while primarily focusing on grounding the unlocalized triplets over image regions. However, they have overlooked the two issues involved in the triplet formation process from the captions: 1) Semantic over-simplification issue arises when extracting triplets from captions, where fine-grained predicates in captions are undesirably converted into coarse-grained predicates, resulting in a long-tailed predicate distribution, and 2) Low-density scene graph issue arises when aligning the triplets in the caption with entity/predicate classes of interest, where many triplets are discarded and not used in training, leading to insufficient supervision. To tackle the two issues, we propose a new approach, i.e., **L**arge **L**anguage **M**odel for weakly-supervised **SGG** (LLM4SGG), where we mitigate the two issues by leveraging the LLM's in-depth understanding of language and reasoning ability during the extraction of triplets from captions and alignment of entity/predicate classes with target data. To further engage the LLM in these processes, we adopt the idea of Chain-of-Thought and the in-context few-shot learning strategy. To validate the effectiveness of LLM4SGG, we conduct extensive experiments on Visual Genome and GQA datasets, showing significant improvements in both Recall@K and mean Recall@K compared to the state-of-the-art WSSGG methods. A further appeal is that LLM4SGG is data-efficient, enabling effective model training with a small amount of training images. Our code is available on https://github.com/rlqja1107/torch-LLM4SGG*

1. Introduction

Scene Graph Generation (SGG) is a fundamental task in computer vision, aiming at extracting structured visual knowledge from images [21, 32, 36, 39, 45, 56]. Most existing SGG methods are fully-supervised, i.e., they heavily rely on the ground-truth annotations that involve the class information of entities and predicates as well as the bounding box of entities [19, 48, 50]. However, since creating extensively annotated scene graph datasets is costly, the heavy reliance on these annotations imposes practical limitations on the model training [54]. To mitigate the high cost associated with manual annotations, weakly-supervised scene graph generation (WSSGG) approaches have recently emerged, aiming at training an SGG model without any annotated scene graph dataset. Specifically, the main idea of recent WSSGG methods is to leverage image captions along with associated images, as they can be easily collected from the Web [20, 47, 54, 57].

The training process of WSSGG model using image captions requires four steps as illustrated in Figure 1(a). **Step 1**: *Preparing an image and its caption.* **Step 2**: *Parsing the image caption,* i.e., triplets formed as ⟨*subject, predicate, object*⟩ are extracted from the image caption through an off-the-shelf parser [30, 43]. **Step 3**: *Aligning the triplets in the caption with entity/predicate classes of interest,* i.e., entity (subject, object) and predicate classes in the extracted triplets obtained in Step 2 are aligned with the entity and predicate classes in the target data[1], respectively. This alignment is based on their synonym/hypernym/hyponym contained in an external knowledge base (KB), e.g., WordNet [25]. **Step 4**: *Grounding unlocalized entities in the extracted triplets,* i.e., unlocalized entities (subjects and objects) are matched with relevant image regions generated by a pre-trained object detector, e.g., Faster R-CNN [29]. The localized entities and predicates in the extracted triplets then serve as pseudo-labels for training an SGG model.

Existing WSSGG approaches mainly focus on Step 4 [20, 33, 47, 57]. For example, in Figure 1(a), their efforts have been focused on grounding the entity person in an unlocalized triplet with an image region that captures the sitting behavior. More precisely, LSWS [47] exploits the con-

*Corresponding Author

[1]We use Visual Genome [13] as the target data.

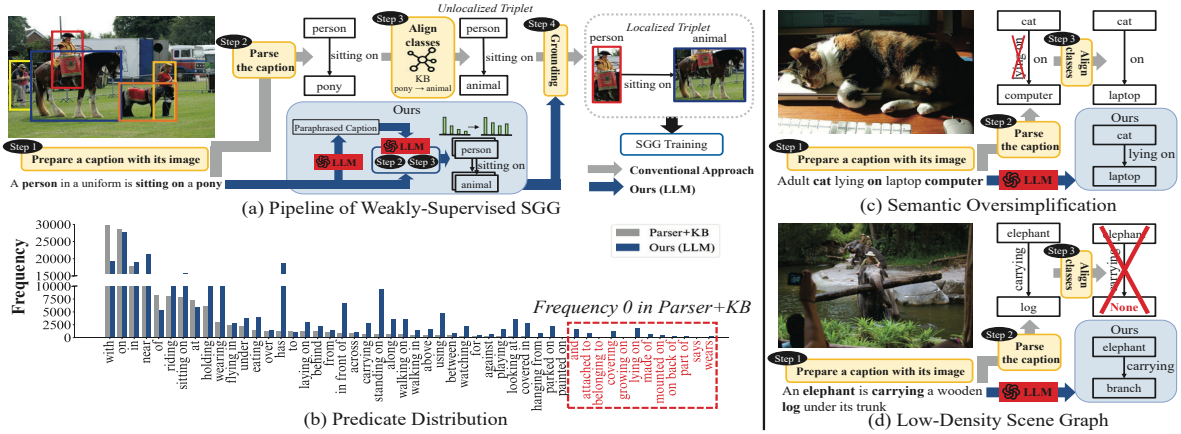

Figure 1. (a) The pipeline of weakly-supervised SGG. (b) The predicate distribution of unlocalized triplets (Parser+KB vs. Ours). In Parser+KB, the distribution becomes heavily long-tailed, and 12 out of 50 predicates are non-existent. (c) Semantic over-simplification caused by a rule-based parser in Step 2. (d) Low-density scene graph caused by the static structure of KB in Step 3.

textual object information to accurately ground the unlocalized entities, leveraging the linguistic structure embedded within the triplets. Another line of research [20] employs a pre-trained vision-language model [15] to reflect the semantic interactions among entities within the image caption.

However, we argue that existing WSSGG approaches overlook the importance of the triplet formation process conducted in Step 2 and Step 3. We identify the two major issues described below, i.e., semantic over-simplication and low-density scene graph, which incur incomplete unlocalized triplets after Step 2 and 3. These incomplete tripletes are mostly uninformative predicates with a limited number, and negatively impact the training of an SGG model even when entities are correctly grounded in Step 4. To demonstrate the impact of incomplete unlocalized triplets, we follow the conventional process to extract unlocalized triplets (i.e., Step 1-3), and conduct an examination of triplets obtained from COCO caption dataset, which are generated through Scene Parser [43] in Step 2 and WordNet [25] in Step 3. As a result, we identify the following two issues:

- **Semantic Over-simplification:** We find that the standard scene graph parser [43] operating on heuristic rule-based principles commonly used in Step 2 leads to a semantic over-simplification of predicates in extracted triplets. In other words, fine-grained predicates are undesirably converted into coarse-grained predicates, which we refer to as semantic over-simplification. For example, in Figure 1(c), an informative predicate lying on (i.e., fine-grained predicate) in the image caption is converted into a less informative predicate on (i.e., coarse-grained predicate), because the rule-based parser fails to capture the predicate lying on at once, and its heuristic rules fall short of accommodating the diverse range of caption's structure. As a result, the predicate distribution becomes heavily long-tailed, in which coarse-grained predicates (e.g., with, on, in) greatly outnumber fine-grained predicates (e.g., parked on, covered in) (Figure 1(b)). To make the matter worse, numerous fine-grained predicates eventually end up in a frequency of 0, even though they are originally present in the captions. Specifically, 12 out of 50 predicates are non-existent, which means that these 12 predicates can never be predicted, since the model is not trained on these predicates at all.
- **Low-Density Scene Graph:** We find that the KB-based triplet alignment in Step 3 leads to low-density scene graphs, i.e., the number of remaining triplets after Step 3 is small. We attribute the low-density scene graphs primarily to the utilization of KB in Step 3. Specifically, a triplet is discarded if any of the three components (i.e., subject, predicate, object) or their synonym/hypernym/hyponym within the triplet fail to align with the entity or predicate classes in the target data. For example, in Figure 1(d), the triplet ⟨elephant, carrying, log⟩ is discarded because log does not exist in Visual Genome dataset nor its synonym/hypernym, even if elephant and carrying do exist. In Table 1, we report the number of triplets and images in Visual Genome dataset, which is a common benchmark dataset used in fully-supervised SGG approaches, and COCO caption dataset, which is a common benchmark dataset used in weakly-supervised SGG approaches. We observe that on average Visual Genome dataset contains 7.1 triplets (i.e., 405K/57K) per image (See Table 1(a)), while COCO dataset contains only 2.4 triplets (i.e., 154K/64K) per image (See Table 1(b)). This indicates that existing WSSGG approaches suffer from the lack of sufficient supervision per image, leading to poor generalization and performance degradation [47, 49]. In summary, relying on

Dataset	How to annotate	# Triplet	# Image
Fully-Supervised approach			
(a) Visual Genome	Manual	405K	57K
Weakly-Supervised approach			
(b) COCO Caption	Parser+KB	154K	64K
(c) COCO Caption	LLM	344K	64K

Table 1. Comparison of scene graph density.

the static structured knowledge of KB is insufficient to cover the semantic relationships among a wide a range of words, which incurs the low-density scene graph after Step 3.

To alleviate the semantic over-simplification and the low-density scene graph issues, we propose a new approach, namely **L**arge **L**anguage **M**odel for weakly-supervised **SGG** (LLM4SGG) that adopts a pre-trained Large Language Model (LLM), which has shown remarkable transferability to various downstream tasks in NLP such as symbolic reasoning, arithmetic, and common-sense reasoning [2, 5, 38]. Inspired by the idea of Chain-of-Thought[2] (CoT) [41], which arrives at an answer in a stepwise manner, we separate the triplet formation process into two chains, each of which replaces the rule-based parser in Step 2 (i.e., Chain-1) and the KB in Step 3 (i.e., Chain-2). More precisely, we design a prompt for extracting triplets from a caption, and ask the LLM to extract triplets formed as <*subject, predicate, object*> (**Chain-1**). We expect that the predicates extracted based on a comprehensive understanding of the caption's context via LLM are semantically rich, thereby alleviating the semantic over-simplification issue. Besides, to alleviate the low-density scene graph issue, we additionally incorporate a paraphrased version of the original caption. To this end, we further design a prompt for paraphrasing the original caption and extracting more triplets from the paraphrased caption. However, entities and predicates in the triplets obtained after Chain-1 are not yet aligned with the target data. Hence, we design a another prompt to align them with entity/predicate classes of interest, and ask the LLM to align them with semantically relevant lexeme included in a predefined lexicon, which is the set of vocabularies that are present in the target data (**Chain-2**). To further engage the LLM in the reasoning process of Chain-1 and Chain-2, we employ the in-context few-shot learning that incorporates a few input-output examples within the prompt, enabling the LLM to perform the task without the need for fine-tuning.

To validate the effectiveness of LLM4SGG, we apply it to state-of-the-art WSSGG methods [54, 57]. Through extensive experiments, we show that LLM4SGG significantly enhances the performance of existing WSSGG methods in terms of mean Recall@K and Recall@K performance on Visual Genome and GQA datasets by alleviating the semantic over-simplification and the low-density scene graph (See Table 1(c) where the number of triplets increased to 334K). A further appeal of LLM4SGG is that it is data-efficient, i.e., it outperforms state-of-the-art baselines even with a small amount of training images, verifying the effectiveness of LLM4SGG.

In summary, we make the following contributions:

- We identify two major issues overlooked by existing WSSGG studies, i.e., semantic over-simplification and low-density scene graph.
- We leverage an LLM along with the CoT strategy and the in-context few-shot learning technique to extract informative triplets without the need for fine-tuning the LLM. To the best of our knowledge, we are the first to leverage an LLM for the SGG task.
- LLM4SGG outperforms the state-of-the-art WSSGG methods, especially in terms of mR@K, demonstrating its efficacy in addressing the long-tail problem in WSSGG for the first time.

2. Related Works

Weakly-Supervised Scene Graph Generation (WSSGG). The WSSGG task aims to train an SGG model without relying on an annotated scene graph dataset. To achieve this, most WSSGG studies [20, 47, 54, 57] utilize image captions and ground unlocalized triplets with image regions. Specifically, VSPNet [49] proposes the iterative graph alignment algorithm to reflect the high-order relations between unlocalized triplets and image regions. SGNLS [57] uses a pre-trained object detector [29] to ground the entities in unlocalized triplets, which share the same classes with the output of object detectors. In addition to the information derived from the object detector, [20] employs a pre-trained vision-language model [15] to capture the semantic interactions among objects. VS^3 [54] uses a grounding-based object detector [18], which calculates the similarity between the entity text in unlocalized triplet and image region, thereby grounding the unlocalized triplets. However, these methods overlook the triplet formation process that leads to the semantic over-simplification (Step 2) and the low-density scene graph (Step 3). In this regard, existing methods result in a sub-optimal performance even when unlocalized triplets are correctly grounded in Step 4.

Large Language Model (LLM). LLMs have demonstrated remarkable transferability to various downstream tasks such as symbolic reasoning, arithmetic, and common-sense reasoning [2, 5, 38]. Specifically, GPT-3 (175B) [2] stands as a cornerstone to break the line of numerous language tasks. Inspired by GPT-3, PaLM (540B) [5], LLaMA (65B) [38], OPT (175B) [53], and LaMDA (137B) [37] have been subsequently introduced. More recently, advanced GPT models (e.g., GPT-4 [28], ChatGPT [27]) fine-tuned with human feedback have gained prominence and widely applied for diverse applications, e.g., planner of tools [9, 24], mobile

[2]We use the CoT strategy as a means to arrive at an answer in a stepwise manner, which differs from the Chain-of-Thought prompting.

task automation [42]. In this work, we employ the power of LLM (i.e., ChatGPT) to alleviate the two issues, i.e., semantic over-simplification and low-density scene graph, in the context of the WSSGG task.

In-Context Few-shot Learning. In-context few-shot learning incorporates a few input-output examples related to a target task, conditioning the LLM on the context of examples. Specifically, GPT-3 [2] pioneered the concept of in-context learning to facilitate an LLM as a versatile model on diverse tasks. This breakthrough has proliferated a plethora of research to leverage the in-context few-shot learning for various tasks [24, 26, 46, 55]. More precisely, Chameleon [24] integrates a few examples to enhance its understanding of tool planning task. [26] utilizes positive and negative examples related to questions for question generation tasks. ReAct [46] incorporates examples of reasoning with action for solving decision-making tasks. Inspired by recent in-context few-shot learning approaches, we provide a few examples to LLMs to help 1) understand the process of triplet extraction from a caption (i.e., Step 2), and 2) align the entity/predicate classes with the target data (i.e., Step 3) in the context of the WSSGG task.

3. Method

In this section, we describe LLM4SGG in detail. We would like to emphasize that LLM4SGG mainly focuses on the triplet formation process conducted in Step 2 (parsing) and Step 3 (aligning), while existing WSSGG approaches mainly focus on Step 4 (grounding). We start by presenting the problem formulation of WSSGG (Section 3.1), followed by the prompt configuration (Section 3.2). Next, we introduce how LLMs are adopted to address the two issues of conventional WSSGG approaches when parsing the image caption (Section 3.3) and aligning the triplets in captions with entity/predicate classes of interest (Section 3.4). Finally, we ground the unlocalized triplets by associating them with bounding boxes (i.e., image regions) and train the SGG model using the localized triplets (Section 3.5). The overall pipeline of LLM4SGG is shown in Figure 2.

3.1. Problem Formulation

In the fully supervised SGG task, we aim to detect a scene graph $\mathbf{G}_f = \{\mathbf{s}_i, \mathbf{p}_i, \mathbf{o}_i\}_{i=1}^{N_f}$ that consists of triplets given an image $\mathcal{I}$, where N_f is the number of triplets in the image. $\mathbf{s}_i$ and $\mathbf{o}_i$ denote the i^{th} subject and the object, respectively, whose bounding boxes are $\mathbf{s}_{i,b}$, $\mathbf{o}_{i,b}$, and entity classes are $\mathbf{s}_{i,c}, \mathbf{o}_{i,c} \in \mathcal{C}_e$, where $\mathcal{C}_e$ is the set of predefined entity classes in the target data. $\mathbf{p}_i$ denotes the predicate between $\mathbf{s}_i$ and $\mathbf{o}_i$, and its class is $\mathbf{p}_{i,c} \in \mathcal{C}_p$, where $\mathcal{C}_p$ is the set of predefined predicate classes in the target data. By using the ground truth scene graphs as supervision, fully supervised SGG approaches train an SGG model $\mathcal{T}_\theta : \mathcal{I} \rightarrow \mathbf{G}_f$, which maps an image to a scene graph.

In the weakly supervised SGG task, we aim to generate a scene graph when the ground truth scene graph is not given, i.e., there are no bounding boxes and entity/predicate class information. Instead, existing WSSGG approaches [20, 33, 47, 54, 57] use image captions along with associated images to produce scene graphs, i.e., localized triplets. More precisely, they extract a set of triplets $\mathbf{G}_w = \{\mathbf{s}_i, \mathbf{p}_i, \mathbf{o}_i\}_{i=1}^{N_w}$ from the image captions, where N_w is the number of triplets extracted from the captions. However, while the extracted triplets contain the class information (i.e., $\mathbf{s}_{i,c}$, $\mathbf{o}_{i,c}$ and $\mathbf{p}_{i,c}$), they are unlocalized, since bounding boxes $\mathbf{s}_{i,b}$ and $\mathbf{o}_{i,b}$ are not included in the caption. Therefore, it is essential to perform the grounding step to associate the unlocalized triplets with bounding boxes. Once we have obtained the localized triplets, we can apply the conventional SGG training scheme, i.e., $\mathcal{T}_\theta : \mathcal{I} \rightarrow \mathbf{G}_f$.

In this paper, our focus is to address the semantic over-simplification and low-density scene graph issues regarding the unlocalized triplets $\mathbf{G}_w$, which has been overlooked in existing WSSGG studies. Specifically, we aim to produce an enhanced $\mathbf{G}_w$ by refining the process of scene graph dataset construction via an LLM. This refinement includes the triplet extraction step from the caption (Step 2) and the alignment of entity/predicate classes (Step 3), leveraging the LLM's comprehensive understanding of language and reasoning ability.

3.2. Prompt Configuration

In fact, it is not a trivial task for an LLM to immediately generate triplets from a caption whose entities and predicates are aligned with entity/predicate classes of interest, as such a task is a novel task for the LLM. Inspired by the idea of the Chain-of-thought (CoT) [41], which arrives at an answer in a stepwise manner, we separate the triplet formation process into the following two chains: Chain-1 – Extracting triplets from captions. Chain-2 – Aligning entities and predicates with the entity/predicate classes of interest. To carefully design each chain, we define the LLM function, i.e., $LLM(\cdot)$, with the following prompt input:

$$Output = LLM(\underbrace{\text{Task description, In-context examples, Actual question}}_{\text{Prompt input}}), \tag{1}$$

where $LLM(\cdot)$ is the decoder of the LLM, generating the $Output$ given the prompt input. The prompt input consists of three components in a sequence: 1) task description, i.e., the delineation of the task that we intend to perform, 2) in-context examples, i.e., sample questions and answers related to the task at hand, 3) actual question, i.e., an inquiry from which we intend to derive the answer. Note that *in-context examples* is closely related to the in-context few-shot learning [2, 40, 41], which is shown to enhance the LLM's understanding of the task. Note that the above configuration of the prompt input is applied to the triplet extraction (Chain-1) (Section 3.3) and the alignment of entity/predicate classes (Chain-2) (Section 3.4).

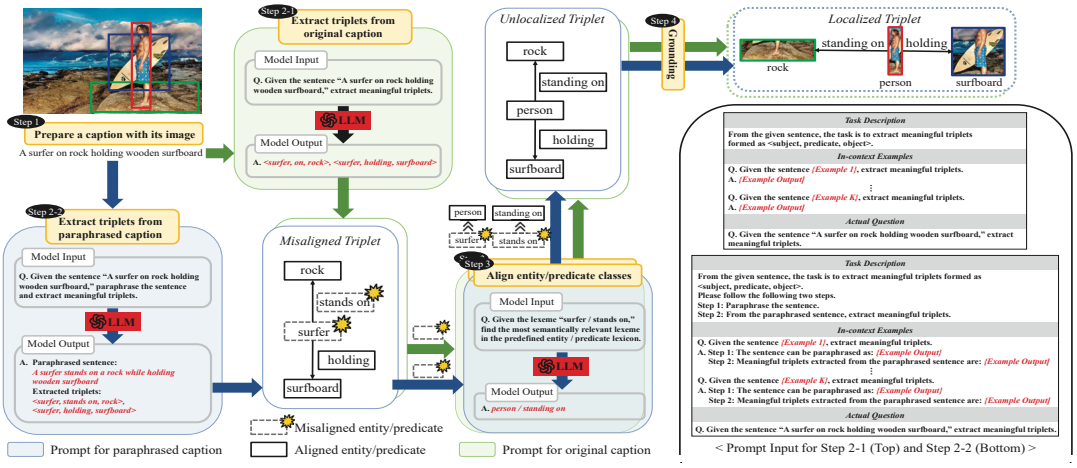

Figure 2. The pipeline of LLM4SGG. Given an image with its caption, we use an LLM to extract triplets from the original caption (Step 2-1) and the paraphrased caption (Step 2-2). Then, we align the entity/predicate classes within the extracted triplets with semantically similar lexeme in the target data via an LLM (Step 3), obtaining the unlocalized triplets. Lastly, we ground the unlocalized triplets over image regions (Step 4) followed by the training of an SGG model.

3.3. Chain-1. Triplet Extraction via LLM (Step 2 in Figure 2)

Based on the LLM's comprehensive understanding of the context of an image caption, we aim to extract triplets from the caption. As discussed in Section 1, we use not only the original caption but also a paraphrased caption generated by the LLM to address the low-density scene graph issue.

Extracting triplets from a *paraphrased* caption (Step 2-2). To extract triplets from a paraphrased caption, we inform the LLM about the task at hand by presenting the following prompt: FROM THE GIVEN SENTENCE, THE TASK IS TO EXTRACT MEANINGFUL TRIPLETS FORMED AS ⟨SUBJECT, PREDICATE, OBJECT⟩ (i.e., Task description in Equation 1). We then instruct the LLM to follow the two steps, i.e., paraphrasing step and triplet extraction step. To help the LLM understand the process of performing the two steps, we present few examples to the LLM that describe how to answer the questions (i.e., In-context examples in Equation 1[3]), which involves a manual construction of questions and corresponding answers to the paraphrasing and the triplet extraction steps. That is, for given a caption, we show the LLM how we expect a paraphrased caption and the extracted triplets would look like (e.g., Given "Four clocks sitting on a floor next to a woman's feet," we show the LLM that a paraphrased sentence would be "Four clocks are placed on the floor beside a woman's feet," and extracted triplets would be ⟨clocks, placed on, floor⟩ and ⟨clocks, beside, feet⟩). Lastly, we show the caption of our interest to the LLM, and let the LLM extract triplets from the caption (i.e., Actual question in Equation 1). Please refer to Figure 2 right (bottom) for an example of the prompt input used in Step 2-2.

[3]We use captions in COCO caption dataset for examples.

Extracting triplets from the *original* caption (Step 2-1). As extracting triplets from the original caption does not involve the caption paraphrasing step, we exclude it from the prompt used in Step 2-2. Please refer to Figure 2 right (top) for an example of the prompt input used in Step 2-1.

In summary, we obtain triplets from both the original and paraphrased captions after Step 2-1 and Step 2-2, respectively, which in turn alleviates the semantic oversimplification issue of predicates and the low-density scene graph issue.

3.4. Chain-2. Alignment of Classes in Triplets via LLM (Step 3 in Figure 2)

The entities (i.e., subject and object) and predicates within the triplets obtained from Step 2 described in Section 3.3 are not yet aligned with the target data. Based on the semantic reasoning ability of the LLM, we aim to align them with the semantically relevant lexeme in the target data.

Aligning entities in the triplets with entity classes of interest. We instruct the LLM with the following prompt: GIVEN THE LEXEME {ENTITY}, FIND SEMANTICALLY RELEVANT LEXEME IN THE PREDEFINED ENTITY LEXICON, where the predefined entity lexicon is $\mathcal{C}_e$ (i.e., Task description in Equation 1). Similar to Section 3.3, we present a few examples to the LLM that describe how to answer the questions (i.e., In-context examples in Equation 1). For example, we provide the LLM with a few examples regarding hierarchical relationships such as pigeon being semantically relevant to bird, and singular-plural relationships such as surfboards being semantically relevant to surfboard. Lastly, we show the entity of our interest to the LLM (i.e., Actual question in Equation 1), which enables the LLM to generate an answer by finding the most

semantically relevant entity in $\mathcal{C}_e$. Please refer to Table 7 in Appendix A.1 for an example of the prompt input.

Aligning predicates in the triplets with predicate classes of interest. Likewise, we instruct the LLM with the following prompt: GIVEN THE LEXEME {PREDICATE}, FIND SEMANTICALLY RELEVANT LEXEME IN THE PREDEFINED PREDICATE LEXICON, where the predefined predicate lexicon is $\mathcal{C}_p$ (i.e., Task description in Equation 1). We also present a few examples to the LLM that describe how to answer the questions (i.e., In-context examples in Equation 1). For example, we provide the LLM with a few examples regarding tense relationships such as the lies on being semantically relevant to lying on, and positional relationship such as next to being semantically relevant to near. Lastly, we show the predicate of our interest to the LLM (i.e., Actual question in Equation 1). Please refer to Table 8 in Appendix A.1 for an example of the prompt input. Furthermore, regarding the alignment of classes within a large predefined lexicon, please refer to Appendix A.2. Note that as our main focus in this work is on developing a framework for utilizing LLMs in weakly-supervised SGG rather than finding a specific prompt design that works the best, we tested with a single prompt design. However, since prompt designs are crucial in leveraging LLMs, we plan to explore various designs in our future work.

After performing Chain-1 (Section 3.3) and Chain-2 (Section 3.4), we obtain intermediate unlocalized triplets $\hat{\mathbf{G}}_w = \{\mathbf{s}_i, \mathbf{p}_i, \mathbf{o}_i\}_{i=1}^{\hat{N}_w}$, where $\mathbf{s}_{i,c}, \mathbf{o}_{i,c} \in \{\mathcal{C}_e \cup \text{None}\}$ and $\mathbf{p}_{i,c} \in \{\mathcal{C}_p \cup \text{None}\}$. It is worth noting that if there is no semantically relevant lexeme, we request the LLM to generate None as the answer, due to the fact that the entity/predicate classes in the target data may not cover a wide range of entities/predicates. Similar to the conventional approach, we discard a triplet if any of its three components (i.e., subject, predicate and object) is None. Lastly, we obtain the final unlocalized triplets $\mathbf{G}_w = \{\mathbf{s}_i, \mathbf{p}_i, \mathbf{o}_i\}_{i=1}^{N_w}$ ($N_w \leq \hat{N}_w$), where $\mathbf{s}_{i,c}, \mathbf{o}_{i,c} \in \mathcal{C}_e$ and $\mathbf{p}_{i,c} \in \mathcal{C}_p$.

3.5. Model Training

Given the final unlocalized triplets $\mathbf{G}_w$, we ground them over relevant image regions to get localized triplets, meaning that we obtain $\mathbf{s}_{i,b}$ and $\mathbf{o}_{i,b}$. To this end, we employ two state-of-the-art grounding methods, i.e., SGNLS [57] and VS^3 [54]. Please refer to Appendix A.3 for more detail about how each method performs grounding. After grounding $\mathbf{G}_w$, we obtain localized triplets and use them as pseudo-labels for training a supervised SGG model. Please refer to Appendix A.4 for more detail about the model training.

4. Experiment

Datasets. To **train** an SGG model without an annotated scene graph dataset, we use three caption datasets: COCO caption [3], Conceptual (CC) caption [31], and Visual Genome (VG) caption [44]. For fair comparisons, we use the same set of images that have been utilized in previous WSSGG studies [20, 47, 54, 57], leading to the utilization of 64K images on COCO caption dataset, 145K images on CC caption dataset, and 57K images on VG caption dataset. To **evaluate** the trained SGG model, we employ the widely used Visual Genome (VG) dataset [13] and GQA dataset [11]. The VG dataset contains the ground-truth localized triplet information annotated by humans. We follow the standard split of VG [44], which consist of 150 entity classes and 50 predicate classes. For the GQA dataset used in the SGG task, we follow the same pre-processing step of a previous SGG study [8], which involves selecting top-200 frequent entity classes and top-100 frequent predicate classes. In both datasets, 30% of the total images are used for evaluation. Please refer to Appendix C.1 for more details regarding the datasets. Note that we mainly use the COCO caption dataset for analysis of LLM4SGG throughout this paper, i.e., CC and VG caption datasets are exclusively used in quantitative result on VG (Section 4.1).

Evaluation metrics. Recent fully-supervised SGG studies [1, 48, 51] have emphasized improving the accuracy of predictions for fine-grained predicates rather than coarse-grained predicates, since the former construct richer scene graphs. As a result, they commonly use mean Recall@K (mR@K) that computes the average of Recall@K (R@K) across all predicates. In line with the recent emphasis on fine-grained predicates, we incorporate both mR@K and R@K in our evaluation, whereas previous WSSGG studies [20, 47, 54] mainly rely on the R@K metric alone. Moreover, we report F@K, which is the harmonic average of R@K and mR@K to jointly consider R@K and mR@K, following previous SGG studies [12, 51]. Regarding the evaluation task, we follow previous WSSGG studies and adopt the Scene Graph Detection (SGDet) task, where both the ground-truth bounding box and the entity class information are not provided. Please refer to Appendix C.2 for more detail regarding the task.

Baselines. Please refer to Appendix C.3 for details regarding the baselines.

Implementation details. Please refer to Appendix C.4 regarding the implementation details.

4.1. Quantitative Result on VG

Table 2 shows the performance of baseline models and those when LLM4SGG is applied. We have the following observations based on COCO caption dataset: **1)** Applying LLM4SGG to SGNLS and VS^3 improves the performance in terms of R@K and mR@K, which demonstrates the effectiveness of the triplet formation through LLM4SGG. Notably, LLM4SGG significantly improves mR@K, implying that LLM4SGG effectively alleviates the long-tailed problem in WSSGG. This can be clearly seen in Figure 3, which shows the performance gain on fine-

Method	Dataset	R@50	R@100	mR@50	mR@100	F@50	F@100
Motif *(CVPR'18)* - Fully-supervised	VG	31.89	36.36	6.38	7.57	10.63 / 12.53	12.53
LSWS *(CVPR'21)*	COCO Caption	3.29	3.69	3.27	3.66	3.28	3.67
SGNLS *(ICCV'21)*		3.80	4.46	2.51	2.78	3.02	3.43
SGNLS *(ICCV'21)*+LLM4SGG		5.09 + 1.29	5.97 + 1.51	4.08 + 1.57	4.49 + 1.71	4.53 + 1.51	5.13 + 1.70
Li et al *(MM'22)*		6.40	7.33	1.73	1.98	2.72	3.12
VS^3 *(CVPR'23)*		6.60	8.01	2.88	3.25	4.01	4.62
VS^3 *(CVPR'23)*+LLM4SGG		**8.91** + 2.31	**10.43** + 2.42	7.11 + 4.23	8.18 + 4.93	**7.91** + 3.90	**9.17** + 4.55
VS^3 *(CVPR'23)*+*Rwt*		4.25	5.04	5.17	5.99	4.67	5.47
VS^3 *(CVPR'23)*+*Rwt*+LLM4SGG		5.10 + 0.85	6.34 + 1.30	**8.42** + 3.25	**9.90** + 3.91	6.35 + 1.69	7.73 + 2.26
VS^3 *(CVPR'23)*	CC Caption	6.69	8.20	1.73	2.04	2.75	3.27
VS^3 *(CVPR'23)*+LLM4SGG		**9.47** + 2.78	**10.69** + 2.49	**5.40** + 3.67	**6.09** + 4.05	**6.88** + 4.13	**7.76** + 4.49
VS^3 *(CVPR'23)*	VG Caption	14.54	18.48	2.80	3.79	4.70	6.29
VS^3 *(CVPR'23)*+LLM4SGG		**18.40** + 3.86	**22.28** + 3.80	**6.26** + 3.46	**7.60** + 3.81	**9.34** + 4.64	**11.33** + 5.04

Table 2. Performance comparisons on the SGDet task. The best performance among WSSGG models within each dataset is in bold. The red numbers indicate the absolute performance improvement after applying LLM4SGG. *Rwt* denotes using the reweighting strategy [51].

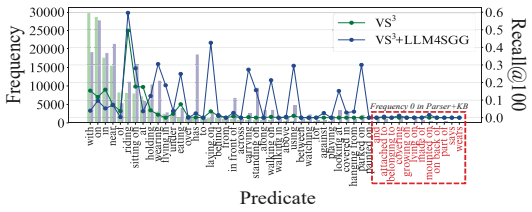

Figure 3. Per class performance (Bar: number of predicate instances, Line: Recall@100).

grained predicates. **2)** VS^3+Rwt+LLM4SGG further improves mR@K of VS^3+Rwt. We attribute this to the fact that the conventional approach generates a limited number of fine-grained predicates, which makes the reweighting strategy less effective within VS^3. Especially, non-existent predicates can never be predicted even when the reweighting strategy is applied. On the other hand, LLM4SGG increases the number of instances that belong to fine-grained predicates, which is advantageous for the reweighting strategy. For the per class performance comparison over the reweighting stregty, please refer to Appendix D.1. **3)** The performance gain obtained from applying LLM4SGG is greater on VS^3 (i.e., VS^3+LLM4SGG) than on SGNLS (i.e., SGNLS+LLM4SGG). The major reason lies in the difference in how SGNLS and VS^3 make use of the pool of 344K unlocalized triplets obtained through LLM4SGG. Specifically, in the grounding process of SGNLS, we observe that 100K out of 344K unlocalized triplets (i.e., 29%) fail to be grounded, and thus not used for training. On the other hand, VS^3 successfully grounds all 344K unlocalized triplets and fully utilize them for training, allowing it to fully enjoy the effectiveness of LLM4SGG. This indicates that LLM4SGG makes synergy when paired with a grounding method that is capable of fully utilizing the unlocalized triplets. For more details regarding the impact of grounding methods, please refer to Appendix B. **4)** Regarding the performance comparison with a fully-supervised approach (i.e., Motif), please refer to Appendix D.2.

Furthermore, we observe that applying LLM4SGG to CC and VG caption datasets also significantly improves performance, demonstrating the effectiveness of LLM4SGG.

Row	PC	LP	LA	# Triplet	R@50 / 100	mR@50 / 100	F@50 / 100
(a)				154K	6.60 / 8.01	2.88 / 3.25	4.01 / 4.62
(b)	✓			243K	9.46 / 11.22	3.43 / 3.92	5.03 / 5.81
(c)	✓	✓		256K	8.42 / 9.85	5.99 / 6.95	7.00 / 8.15
(d)	✓		✓	327K	**11.76 / 13.38**	3.50 / 4.05	5.39 / 6.22
(e)	✓	✓	✓	344K	8.91 / 10.43	**7.11 / 8.18**	**7.91 / 9.17**

Table 3. Ablation studies. (PC: Using Paraphrased Caption in addition to the original caption / LP: LLM-based Parsing / LA: LLM-based Alignment)

4.2. Ablation Studies

In Table 3, we conduct ablation studies on VG dataset to understand the effectiveness of each component of LLM4SGG, where VS^3 is used as the grounding method. Note that row (a) is equivalent to vanilla VS^3. We have the following observations. **1) Effect of using the paraphrased caption:** Including the paraphrased caption in addition to the original caption (row (b)) increases the number of triplets (154K→243K), resulting in an improved overall performance. This demonstrates that the paraphrased caption alleviates the low-density scene graph issue. **2) Effect of LLM-based parsing:** The LLM-based parsing (row (c)) for extracting triplets improves mR@K of row (b). This indicates that the LLM-based parsing increases the number of instances that belong to fine-grained predicates, which in turn alleviates the semantic over-simplification issue. **3) Effect of LLM-based alignment:** The LLM-based alignment (row (d)) of entities/predicates in the extracted triplets increases the number of triplets from 243K to 327K (row (b) vs (d)), which indicates that the low-density scene graph issue is alleviated. Consequently, R@K and mR@K of row (d) are greater than those of row (b). **4)** The fully-fledged approach (row (e)) generally improves R@K and mR@K, showing the best performance in terms of F@K. It is important to highlight that when using the LLM-based parsing, the performance of mR@K significantly increases with a moderate decrease in R@K. This trade-off is attributed to the fact that R@K generally improves when the coarse-grained predicates are dominant [51]. In contrast, our approach, which addresses the semantic over-simplification issue, decreases the number instances that belong to coarse-grained predicates while simultaneously increasing those that belong to fine-grained predicates (Figure 1(b)), which

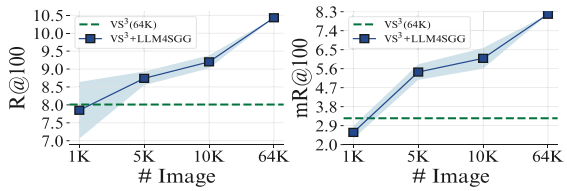

Figure 4. Performance over various numbers of images used for training VS^3+LLM4SGG.

in turn results in a substantial improvement in mR@K. We would like to emphasize that the performance in terms of mR@K is crucial in the context of SGG research [1, 8, 48], as fine-grained predicates offer richer information.

4.3. Analysis of Data-Efficiency

To assess the effectiveness of LLM4SGG under the lack of available training data, we conduct experiments given a limited number of images. Specifically, among 64K images used for training VS^3 and VS^3+LLM4SGG in Table 2, we randomly sample 1K (1.5%), 5K (7.8%), and 10K (15.6%) images with replacement, and train VS^3+LLM4SGG for five times. Figure 4 shows the average performance over various numbers of images used for training VS^3+LLM4SGG along with the variance (in blue area). We observe that when using only 1K images, the performance is slightly inferior to the baseline that used 64K for training (i.e., VS^3). However, as we increase the number of images used for training to 5K images, we observe a significant improvement in both R@K and mR@K compared with the baseline. This demonstrates that LLM4SGG is data-efficient, as *it outperforms the baseline even with only 7.8% of the total images used for training the baseline*. Moreover, when further increasing the number of images to 10K, we observe further performance improvements, and when it reaches 64K, which is the same as the number of images used for training the baseline, the performance is the best. In summary, LLM4SGG enables data-efficient model training even with a limited amount of available images for training, thanks to alleviating the semantic over-simplification and low-density scene graph issues.

4.4. Quantitative Result on GQA

In Table 4, we additionally conducted experiments on GQA dataset [11]. Please refer to Appendix E.1 for more detailed descriptions on the training and evaluation processes on GQA dataset. The GQA dataset contains twice as many predicates as the Visual Genome dataset and includes complicated predicates (e.g., sitting next to, standing in front of). As a result, when obtaining unlocalized triplets using the conventional WSSGG approach, we observe that 44 out of 100 predicates have a frequency of 0, and the predicate distribution is extremely long-tailed. Consequently, the baseline (i.e., VS^3) exhibits significantly lower performance, especially in terms of mR@K. On the other hand,

Method	R@50 / 100	mR@50 / 100	F@50 / 100
Motif (Fully-supervised)	28.90 / 33.10	6.40 / 7.70	10.48 / 12.49
VS^3	5.90 / 6.97	1.60 / 1.81	2.52 / 2.87
VS^3+LLM4SGG	**8.88 / 10.38**	**5.33 / 6.51**	**6.66 / 8.00**

Table 4. Performance comparison on GQA.

our approach shows substantial performance improvements not only in R@K but also in mR@K, thanks to the mitigation of semantic over-simplification and low-density scene graph issues. Please refer to Appendix E.2 for the predicate distribution and performance comparison for each class in GQA dataset. Additionally, please refer to Appendix E.3 for qualitative analyses on GQA dataset.

5. Conclusion & Future work

In this work, we focus on the triplet formation process in WSSGG, whose importance is overlooked by previous studies. To alleviate the semantic over-simplification and low-density scene graph issues inherent in the triplet formation process, we propose a new approach, i.e., LLM4SGG, which leverages a pre-trained LLM during the extraction of triplets from the captions, and alignment of entity/predicate classes with those in the target data. It is important to note that construction of these triplets is not required every time to train the SGG model; instead, it is a one-time pre-processing step. In this regard, we contribute to generating enhanced triplets compared to the conventional approach. Moreover, we publish an enhanced SGG dataset constructed by LLM for future studies of SGG. As a result, we outperform baselines in terms of R@K, mR@K and F@K on Visual Genome and GQA datasets. A potential limitation of our work is the reliance on a proprietary blackbox LLM, which can also be costly to use. Hence, in Appendix F, we provide discussions on replacing the LLM with smaller language models.

For future work, an LLM can be used to ground the unlocalized triplets in Step 4. Recently, vision-language representation learning has been developed for transforming visual features into textual features to facilitate the use of visual features as input to an LLM [16, 58]. In this regard, given the visual features of bounding boxes as input, we could ask the LLM to identify relevant bounding boxes based on the textual information of entities within unlocalized triplets using the comprehensive understanding of the context of triplets.

6. Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2020-0-00004, Development of Previsional Intelligence based on Long-term Visual Memory Network, No.2022-0-00077, Reasoning, and Inference from Heterogeneous Data, No. 2019-0-00079, Artificial Intelligence Graduate School Program, Korea University).

References

[1] Bashirul Azam Biswas and Qiang Ji. Probabilistic debiasing of scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10429–10438, 2023. 6, 8

[2] 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. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3, 4

[3] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 6

[4] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer, 2020. 2, 3

[5] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. 3

[6] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7373–7382, 2021. 2, 3, 7

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 3, 7

[8] Xingning Dong, Tian Gan, Xuemeng Song, Jianlong Wu, Yuan Cheng, and Liqiang Nie. Stacked hybrid-attention and group collaborative learning for unbiased scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19427–19436, 2022. 6, 8, 3

[9] Difei Gao, Lei Ji, Luowei Zhou, Kevin Qinghong Lin, Joya Chen, Zihan Fan, and Mike Zheng Shou. Assistgpt: A general multi-modal assistant that can plan, execute, inspect, and learn. *arXiv preprint arXiv:2306.08640*, 2023. 3

[10] Tao He, Lianli Gao, Jingkuan Song, and Yuan-Fang Li. Towards open-vocabulary scene graph generation with prompt-based finetuning. In *European Conference on Computer Vision*, pages 56–73. Springer, 2022. 3

[11] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019. 6, 8, 3, 7

[12] Siddhesh Khandelwal and Leonid Sigal. Iterative scene graph generation. *Advances in Neural Information Processing Systems*, 35:24295–24308, 2022. 6, 3

[13] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017. 1, 6, 2, 3, 5, 7

[14] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. 2, 3, 4

[15] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021. 2, 3, 4

[16] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 8

[17] Lin Li, Guikun Chen, Jun Xiao, Yi Yang, Chunping Wang, and Long Chen. Compositional feature augmentation for unbiased scene graph generation. *arXiv preprint arXiv:2308.06712*, 2023. 3

[18] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022. 3, 2, 4

[19] Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11109–11119, 2021. 1, 3

[20] Xingchen Li, Long Chen, Wenbo Ma, Yi Yang, and Jun Xiao. Integrating object-aware and interaction-aware knowledge for weakly supervised scene graph generation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4204–4213, 2022. 1, 2, 3, 4, 6

[21] Yongzhi Li, Duo Zhang, and Yadong Mu. Visual-semantic matching by exploring high-order attention and distraction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12786–12795, 2020. 1

[22] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 4

[24] Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. *arXiv preprint arXiv:2304.09842*, 2023. 3, 4

[25] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 1, 2

[26] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773*, 2021. 4

[27] OpenAI. Chatgpt. *https://openai.com/blog/chatgpt*, 2023. 3, 4

[28] OpenAI. Gpt-4. Technical report, 2023. 3

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 1, 3, 2, 4

[30] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, pages 70–80, 2015. 1

[31] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018. 6

[32] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8376–8384, 2019. 1

[33] Jing Shi, Yiwu Zhong, Ning Xu, Yin Li, and Chenliang Xu. A simple baseline for weakly-supervised scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16393–16402, 2021. 1, 4

[34] Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. Energy-based learning for scene graph generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13936–13945, 2021. 3

[35] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3716–3725, 2020. 5

[36] Damien Teney, Lingqiao Liu, and Anton van Den Hengel. Graph-structured representations for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2017. 1

[37] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022. 3

[38] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 3, 9

[39] Sijin Wang, Ruiping Wang, Ziwei Yao, Shiguang Shan, and Xilin Chen. Cross-modal scene graph matching for relationship-aware image-text retrieval. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1508–1517, 2020. 1

[40] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022. 4

[41] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022. 3, 4, 6

[42] Hao Wen, Yuanchun Li, Guohong Liu, Shanhui Zhao, Tao Yu, Toby Jia-Jun Li, Shiqi Jiang, Yunhao Liu, Yaqin Zhang, and Yunxin Liu. Empowering llm to use smartphone for intelligent task automation. *arXiv preprint arXiv:2308.15272*, 2023. 4

[43] Hao Wu, Jiayuan Mao, Yufeng Zhang, Yuning Jiang, Lei Li, Weiwei Sun, and Wei-Ying Ma. Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6609–6618, 2019. 1, 2, 6, 8

[44] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5419, 2017. 6

[45] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10685–10694, 2019. 1

[46] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022. 4

[47] Keren Ye and Adriana Kovashka. Linguistic structures as weak supervision for visual scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8289–8299, 2021. 1, 2, 3, 4, 6

[48] Kanghoon Yoon, Kibum Kim, Jinyoung Moon, and Chanyoung Park. Unbiased heterogeneous scene graph generation with relation-aware message passing neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3285–3294, 2023. 1, 6, 8, 3

[49] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Weakly supervised visual semantic parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3736–3745, 2020. 2, 3

[50] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5831–5840, 2018. 1, 3, 4

[51] Ao Zhang, Yuan Yao, Qianyu Chen, Wei Ji, Zhiyuan Liu, Maosong Sun, and Tat-Seng Chua. Fine-grained scene graph generation with data transfer. In *European conference on computer vision*, pages 409–424. Springer, 2022. 6, 7, 3, 5

[52] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5579–5588, 2021. 1

[53] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 3

[54] Yong Zhang, Yingwei Pan, Ting Yao, Rui Huang, Tao Mei, and Chang-Wen Chen. Learning to generate language-supervised and open-vocabulary scene graph using pre-trained visual-semantic space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2915–2924, 2023. 1, 3, 4, 6, 2, 7, 8

[55] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*, 2022. 4

[56] Yiwu Zhong, Liwei Wang, Jianshu Chen, Dong Yu, and Yin Li. Comprehensive image captioning via scene graph decomposition. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 211–229. Springer, 2020. 1

[57] Yiwu Zhong, Jing Shi, Jianwei Yang, Chenliang Xu, and Yin Li. Learning to generate scene graph from natural language supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1823–1834, 2021. 1, 3, 4, 6, 2

[58] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 8