CLIP-Prefix for Image Captioning and an Experiment on Blind Image Guessing

Triet Minh Huynh[1], Duy Linh Nguyen[1], Thanh Tri Nguyen[1], Thuy-Duong Thi Vu[1], Hanh Dang-Ngoc[2,3], and Duc Ngoc Minh Dang[1(✉)]

[1] Computing Fundamental Department, FPT University, Ho Chi Minh City, Vietnam
{triethmse160251,linhndse150400,trintse161160}@fpt.edu.vn, {duongvtt9,ducdnm2}@fe.edu.vn

[2] Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam
hanhdn@hcmut.edu.vn

[3] Vietnam National University Ho Chi Minh City, Ho Chi Minh City, Vietnam

Abstract. Image caption generation resides at the intersection of computer vision and natural language processing, with its primary goal being the creation of descriptive and coherent textual narratives that faithfully depict the content of an image. This paper presents two models that leverage CLIP as the image encoder and fine-tune GPT-2 for caption generation on the Flickr30k and Flickr8k datasets. The first model utilizes a straightforward mapping network and outperforms the original architecture with a BLEU-1 score of 0.700, BLEU-4 score of 0.257, and ROUGE score of 0.569 on the Flickr8k dataset. The second model constitutes a new architecture exploring the boundaries of minimal visual information required for captioning. It incorporates CLIP's text encoder to produce input for the generator, while the image embedding serves solely as a validation mechanism. Despite its relatively lower performance, with a BLEU-1 score of 0.546, BLEU-4 score of 0.108, and ROUGE score of 0.444 on the Flickr8k dataset, this model demonstrates the decoder's ability to create captions based on keyword descriptions alone, without direct access to the context vector.

Keywords: GPT-2 · image caption generation · CLIP · OpenCLIP · sentence transformer · zero-shot text classification

1 Introduction

Image captioning presents significant challenges, demanding the generation of meaningful and contextually relevant captions in natural language for given input images. Within this domain, two key hurdles emerge: semantic understanding

N.-S. Vo et al. (Eds.): INISCOM 2024, LNICST 595, pp. 189–203, 2024.
https://doi.org/10.1007/978-3-031-67357-3_14

and the myriad ways a single image can be described. Semantic understanding involves identifying the primary object and comprehending relationships between different elements within the image. In addressing these challenges, this paper draws inspiration from recent advancements in vision-language pretraining models, explicitly leveraging the capabilities of Contrastive Language-Image Pre-Training (CLIP) [1].

1.1 Image Captioning

In image captioning, models traditionally learn from annotated image-caption pairs, optimizing for the accurate generation of captions. The prevalent architecture involves an encoder-decoder framework [2–4], where an image encoder is paired with a language model. Attention-based models [5–8] have gained prominence, integrating visual attention mechanisms [9] to enhance interpretability and informativeness in generated captions. These models prioritize specific image regions, dynamically adjusting attention weights based on contextual relevance. Exploring alternative approaches, generative adversarial networks (GANs) [10] have been applied to image captioning. GAN-based models [11,12] utilize a generator-discriminator architecture to enhance caption quality, striving for a faithful representation of visual content. Furthermore, reinforcement learning (RL) has found its place in image captioning frameworks [13], enabling models to learn optimal captioning policies. RL models often incorporate reward mechanisms to reinforce caption generation aligned with human preferences and linguistic norms. In contrast, image-captioning research using unsupervised learning methods has recently gained traction. In such cases, the challenge lies in aligning vision and language without explicit annotations. Successful approaches in [14,15] involve training caption generators to comprehend human language and address image-text disparities. The diversity within the image captioning landscape thus spans traditional annotated data-driven architectures to innovative approaches tackling the alignment challenge without explicit annotations.

1.2 CLIP-Based Image Captioning

Within image captioning, CLIP-based methodologies have undergone extensive exploration, particularly in scenarios with paired training data. Prior research endeavors leveraging CLIP as a backbone [16,17] have demonstrated notable improvements in captioning performance. CLIP introduces a shared representation for images and text prompts, achieved through extensive training on a diverse set of images and textual descriptions using a contrastive loss. This shared representation empowers the model to effectively capture semantic relationships, thereby streamlining both training time and data requirements.

1.3 Our Approaches

The first of two architectures we are developing draws inspiration from CLIP-based research. In this CLIP-prefix approach, a mapping network is employed

to generate a prefixed sequence for each caption, utilizing the CLIP embedding. This fixed-size prefix is then concatenated with the caption embedding and fed into a language model, specifically GPT-2 [18], a model previously validated for its capacity to produce varied and comprehensive textual outputs. Simultaneously, the language model undergoes fine-tuning concurrently with the training of the mapping network. We introduce multiple variants of this architecture, incorporating gradient clipping to control overfitting and re-train the language model's tokenizer to better adapt to the dataset's vocabulary. During the inference phase, the language model incrementally generates the caption, starting from the prefix. However, the foundational concept of this approach, akin to various strategies employing an encoder-decoder framework [2–4], is rooted in the notion of bridging the gap between the visual and textual domains with the neural network. In this case, the visual embedding serves as the context vector, allowing the decoder to 'see' the content depicted in the image. The overarching objective of most researchers engaged in developing image-caption models is to extract maximal information from this context vector.

In contrast, for our second approach, we propose Selective Blind Guessing (SBG), which adopts an alternative perspective aiming to delineate the minimal visual information required for captioning. Leveraging CLIP's text encoder and its contrastive learning capability enables the association of semantic relevance to text based on cosine similarity, which transforms the image embedding into a list of keywords. Thereby foregoing the context vector and any potential minor information that could have been utilized for a more nuanced understanding of the image, effectively 'blinding' the model.

Our SBG strategy seeks to subvert the established encoder-decoder paradigm with a rule-based probabilistic approach, prioritizing the use of CLIP's text encoder as the primary conduit for cross-modal visual comprehension. In this approach, the image encoder assumes a supplementary role as a corroborative agent tasked with validating semantic information.

It is imperative to highlight that the CLIP used in our models is specifically OpenCLIP [19], an open-source implementation of the proprietary CLIP. This choice of model opens up the prospect of conducting tests involving diverse pre-trained configurations in future investigations. Nevertheless, for this experiment, we employ weights identical to those of the original CLIP model, negating any differences.

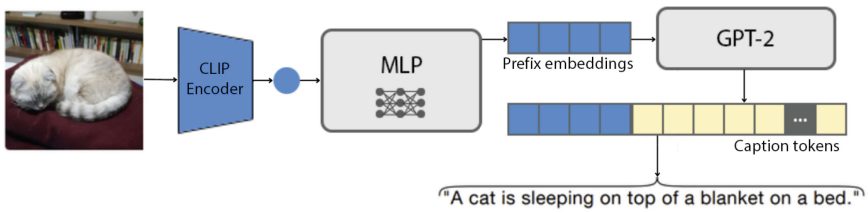

Fig. 1. The CLIP-prefix architecture.

2 Methodology

2.1 Problem Statement

Given a dataset of image-caption pairs $\{x_i, c_i\}_{i=1}^{N}$, where each image x_i is encoded as a feature using CLIP's image encoder, denoted as $X_i = \text{ICLIP}(x_i)$. The caption is a sequence of tokens $c_i = c_i^1, ..., c_i^n$. Our goal for both architectures is to learn the mapping function $\mathcal{F} : X \rightarrow c$.

2.2 CLIP-Prefix Mapping Network

Built upon [16], as illustrated in Fig. 1, this model utilizes a mapping network $\mathcal{M}$, which is a simple multi-layer perception (MLP) to convert the image embedding to a sequence of k embedding vectors:

$$p_i^1, ..., p_i^k = \mathcal{M}(X_i), \tag{1}$$

where each vector p_i^j has the same dimension as the word embedding. We then combine it with the caption tokens $c_i^1, ..., c_i^l$ where l is the maximal length that we pad our tokens to obtain the final prefix-caption concatenation:

$$Z_i = p_i^1, ..., p_i^k, c_i^1, ..., c_i^l. \tag{2}$$

During the training process, the concatenation is provided as input to the GPT-2 language model. Given the prefix, the primary goal is to predict caption tokens autoregressively. This predictive task is executed through the use of a straightforward cross-entropy loss:

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{j=1}^{l} \log p_\theta(c_i^j \mid p_i^1, ..., p_i^k, c_i^1, ..., c_i^{j-1}). \tag{3}$$

For the alterations applied to the original model, we now introduce two variants, each tackling one field of improvement.

Gradient Clipping. Gradient clipping is a regularization technique designed to address the issue of exploding gradients during training. This phenomenon occurs when the gradients of the loss function become excessively large, leading to unstable learning dynamics and impeding convergence. By imposing a threshold on the gradients during backpropagation, gradient clipping mitigates the risk of such explosive behavior. This process involves computing the gradients of the cross-entropy loss function with respect to the model parameters, denoted as $\nabla_\theta \mathcal{L}$, and applying clipping as follows:

$$\hat{g}_\tau = \text{clip}_\tau(\nabla_\theta \mathcal{L}), \tag{4}$$

where τ is a predefined threshold determining the maximum absolute value allowed for the gradients, the clipped gradients, denoted as $\hat{g}$, serve as the basis for updating the model parameters to achieve more stable training:

$$\theta_{\text{new}} = \theta_{\text{old}} - \eta \hat{g}_\tau. \tag{5}$$

The decision to integrate gradient clipping stems from a proactive strategy to mitigate overfitting tendencies associated with the repeating extensive training of the dataset. This approach acts as a regularization method, ensuring that the model does not overly tailor itself to the intricacies of the training data, thereby promoting more controlled learning dynamics.

Custom Tokenizer. The motivation behind a custom tokenizer stems from the recognition that the inherent characteristics and linguistic patterns present in the captions may not be optimally captured by a generic off-the-shelf tokenizer. Therefore, a new tokenizer is trained on each dataset's captions to discern and appropriately handle domain-specific terminology and contextual nuances. Leveraging the GPT-2's subword tokenization with the same vocabulary size for easy comparison, we adapt the tokenizer to encode text using fewer tokens, enhancing efficiency while preserving semantic richness.

2.3 Selective Blind Guessing

Inspiration and Conceptual Framework: The genesis of the SBG model originated from the widely acknowledged "20 Questions" game. Traditionally, this recreational activity involves two participants, namely the "questioner" (denoted as Q) and the "answerer" (A). During the game, the answerer selects a final subject $\mathcal{S}$ for the questioner to deduce, while the latter is allotted a maximum of 20 queries. Notably, the responses $R_j = A(Q_j)$ to these queries are binary, confined to either *yes* or *no*, with pairwise relation to the question. The strategic essence of the game lies in the systematic progression from overarching conceptual domains towards more specific inquiries, depending upon the received responses:

$$\mathcal{S} = \left\{ \prod_{j=1}^{m} Q_j \longrightarrow R_j \;\middle|\; \substack{1 \leq m \leq 20 \\ R_j \in \{yes, no\}} \right\}. \tag{6}$$

As for our model, each query is a list of keywords K_j, and the answerer is CLIP's image encoding X_i, responding with:

$$R_j = r_j^1, ..., r_j^n = \textbf{Similarity}(X_i, \text{TCLIP}(K_j)), \tag{7}$$

where TCLIP is CLIP's text encoder and R_j is a list of probabilities corresponding to each keyword in K_j, based on cosine similarity and normalized to a probability distribution via softmax. The entire process of caption prediction per image would be as follows:

$$\sum_{u=1}^{m_1} \mathcal{S}_u^i = \left\{ \sum_{u=1}^{m_1} \prod_{j=1}^{m_2} (K_j(\text{argmax}(R_j))_u \;\middle|\; \substack{m_1 \times m_2 = m \\ K_j \longrightarrow R_j} \right\}, \tag{8}$$

$$c_i = \mathcal{F}\left(\sum_{u=1}^{m_1} \mathcal{S}_u^i \right). \tag{9}$$

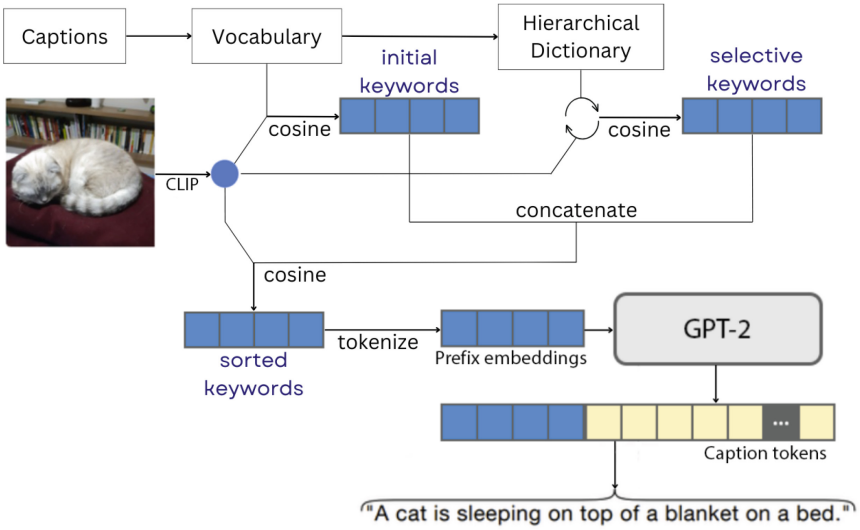

Fig. 2. The Selective Blind Guessing architecture.

In this context, the foundational principle remains constant. However, the scope extends beyond identifying a single subject as an image inherently encompasses multiple. The iterative process persists, constrained by a predetermined limit of m questions where m_1 represents the count of subjects to identify and m_2 denotes the number of queries performed for each subject. After each query, the keyword of the highest probability in the K_j list is selected. Throughout this process, the role of image embedding is 'pacified' to a supporting function solely for validating the image's content, as seen in Fig. 2, and at no point is actively utilized in the generation task. Next, we will delve into the process of vocabulary creation and query extraction.

Vocabulary and First Query Extraction: Before submitting queries to our model, it is imperative to formulate a lexicon from which we can derive our keywords. This involves employing the lemmatization process on the complete textual corpus, transforming words into their fundamental forms according to their part-of-speech tags. The initial vocabulary comprises all distinct words organized in descending order based on their respective frequencies. Moreover, only some words are essential; some contribute minimally to the textual insight, while others are entirely superfluous. Therefore, we only include those that fall into the categories of Noun, Verb, Adjective, Adverb, or Number. Despite the meticulous construction of our initial vocabulary, it becomes apparent that including synonyms and semantically similar words may lead to redundancy and potentially dilute our extraction. To refine our lexicon further, we leverage a sentence transformer (SBERT) [20], which employs sentence embedding (in this case, word embedding) to capture their contextual meaning. It enables us to compare words

based on their semantic similarity to one another through cosine similarity:

$$scores_i = \text{SBERT}_\tau(w_i \mid w_{i+1}, ..., w_n). \tag{10}$$

Unlike other embedding methods, which focus on unsupervised learning to detect semantic patterns, SBERT incorporates triplet networking [21]. This approach aids in distinguishing synonyms from those of opposite meanings but with similar contextual associations. It allows us to identify and filter out synonyms with similarity exceeding the threshold τ over earlier instances in our lexicon. This process ensures that our final vocabulary is not only comprehensive but also devoid of unnecessary duplication.

Now, we can begin our first query. Despite the substantial reduction in lexicon size through the filtration steps, encoding the entire set would be computationally inefficient. Thus, we opt to encode only the top 1000 most frequent of them (excluding those categorized as Number) for our question:

$$K_1 = w_1, ..., w_{1000}. \tag{11}$$

After obtaining the probability list $R_1 = r_1, ..., r_{1000}$, the subsequent task involves selecting the most relevant keywords to form our first answer $\mathcal{S}_1$. This is achieved by sorting the K_1 list in descending order based on R_1 values and extracting the first l words with which the cumulative sum of probability reaches a threshold τ:

$$\mathcal{S}_1 = \left\{ w_1, ..., w_l \;\middle|\; \sum_{j=1}^{l} r_j = \tau \right\}. \tag{12}$$

While this approach excels at extracting a diverse array of relevant keywords, its broad nature results in overlooking specific subjects emphasized within the captions. Therefore, we transition to the main method of keyword extraction.

Selective Query Extraction: This approach closely follows the principles behind the "20 Questions" game, aiming to strategically navigate from a broad initial concept to pinpoint primary subjects, refining our selection iteratively. In implementing this strategy, we construct a hierarchical dictionary with a two-level depth, as shown in Fig. 3. In the first layer, we capture the broadest conceptual categories with keys that represent overarching themes spanning a spectrum of potential subjects.

Moving to the second layer, our focus sharpens on each concept. Here, the keys offer a more granular classification for every theme identified in the first layer. Each key is linked to a list containing the actual keywords from the vocabulary. It is worth noting that not all keywords align with the two-level traversal strategy, as shown in Fig. 3. While some are readily identifiable and can be directly retrieved from the first layer, others exhibit inter-layer dependencies, necessitating additional retrievals. This flexibility ensures a tailored classification system. To populate our hierarchical dictionary (denoted as $\mathcal{D}$) with keywords, we employ a zero-shot text classifier [22]. This method involves discerning the

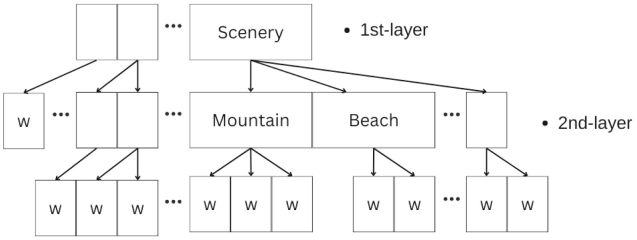

Fig. 3. The hierarchical dictionary.

logical relationship between a keyword w_i and potential categories $\mathcal{C}$ without explicit training. In the framework of our classifier, this methodology facilitates precise classification by ascribing similarity scores to categories and selecting the one with the highest score according to the following relation:

$$w_i \in \mathcal{D}_{key}, \quad \text{where} \quad key = \mathcal{C}(\text{argmax}(\text{Classifier}(w_i \mid \mathcal{C}))). \tag{13}$$

With this, we can retrieve the remaining keywords through the selective process of traversing the dictionary:

$$\sum_{u=2}^{m_1} \mathcal{S}_u = \sum_{u=2}^{m_1} \prod_{j=1}^{m_2} (\mathcal{D}_{key_j}(\text{argmax}(R_j))_u, \tag{14}$$

where $\mathcal{D}_{key_j}$ signifies the set of available values with key_j being the result from the previous query of each subject. As for the first query of each subject, key would be the subject's broadest domain, which, alongside the count m_1, can be identified by taking combinations of the dictionary's first-level keys as a special query at the start of this procedure and select the one with the highest probability. The biggest advantage of this method lies in its capability to pinpoint predefined subjects within an image while making optimal use of the entire vocabulary. Nevertheless, there is a caveat to this tactic. If it erroneously assigns the image to the wrong classes, it would go down the wrong path to selecting unrelated words of local highest probability, thereby compromising the quality of our selection. To address this issue, we concatenate all subjects to compose the final list of keywords and sort its results in descending order of probability.

$$\mathcal{S}_{final} = sort\left(\sum_{u=1}^{m_1} \mathcal{S}_u,\ R_{final}\right). \tag{15}$$

This ensures that irrelevant words are relegated to the end of the list, which is beneficial since, during training, the generator (GPT-2) will learn that the beginning of the sequence contains the most relevant keywords. Regarding the training procedure, we would once again take the cross-entropy loss, with the tokenized result of the final query as the prefix:

$$\mathcal{L} = -\sum_{i=1}^{N} \sum_{j=1}^{l} \log p_\theta(c_i^j \mid \mathcal{S}_i^{final}, c_i^1, ..., c_i^{j-1}). \tag{16}$$

Inference: In the inference phase, we devise two types of variation for our SBG architecture. One is a standard version that generates a single caption. In contrast, the others generate multiple captions and re-employ CLIP's querying process with those captions as respective keywords to identify the one with the highest probability.

3 Performance Evaluation

3.1 Experiment Setup

Our image caption models were trained and evaluated on two prominent benchmark datasets, namely Flickr8k [23] and Flickr30k [24]. The Flickr8k dataset is a widely recognized benchmark in the field of image captioning. It comprises over 8,000 high-quality images, each accompanied by five human-generated captions. The images encompass diverse scenes, objects, and activities, making them a valuable resource for assessing the model's ability to comprehend various visual content. In addition to Flickr8k, we leveraged the Flickr30k dataset to provide a more expansive evaluation environment. Consisting of 31,000 images, including ones from Flickr8k, this dataset surpasses its predecessor in size and complexity with a more extensive vocabulary and longer caption sequences. Each dataset underwent partitioning, wherein 1000 images were allocated for testing, another 1000 for validation, and the remaining subset for training. This partitioning strategy was deliberately implemented to facilitate a systematic comparison with benchmark outcomes derived from models trained on such datasets using the Karpathy split [2], which adheres to a similar distribution ratio. Regarding preprocessing, the image data remains unchanged. CLIP was trained on a substantial corpus of 400 million image-caption pairs and is already proficient in extracting features from a diverse array of images. On the other hand, the caption data for both models undertook a uniform conversion process to lowercase. In the case of the SBG model, an additional step was taken to construct a vocabulary from which keywords were extracted. This vocabulary of lemmatization contains 5346 words for the Flickr8k dataset and 10735 words for the Flickr30k dataset.

To assess the efficacy of our models, we choose a set of well-established evaluation metrics encompassing a range of linguistic and content-based criteria. Firstly, we utilize the BLEU (Bilingual Evaluation Understudy) metric, specifically BLEU-1 to BLEU-4, which measures the precision of n-gram overlap between the generated captions and reference captions. The ROUGE-L metric is also employed, focusing on the longest common subsequence between the generated and reference captions. Furthermore, the METEOR (Metric for Evaluation of Translation with Explicit Ordering) metric is applied, offering a comprehensive evaluation that considers precision, recall, and stemming. METEOR assesses not only lexical overlap but also incorporates semantic considerations.

In terms of baselines, we conducted a comparative analysis between our two models and their respective variants against the original CLIP-prefix model [16]. This model encodes images with CLIP and incorporates a simple MLP as the

mapping network. We also compared against the benchmark results of the paper [4], which uses the same datasets, albeit with the Karpathy split. This paper explores the application of a recurrent neural network (RNN) as a text encoder, subsequently merging it with image features to generate captions. We opted for the 256-layer version with full vocabulary size (3.478 for Flickr8k and 9.584 for Flickr30k), as it demonstrates the highest performance on Flickr8k and satisfactory performance on Flickr30k.

3.2 Performance Results

From the result of Tables 1, 2 and 3, we can see a general trend in that models trained on the Flickr8k dataset tend to perform better than those of Flickr30k. This is due to the reduction in vocabulary and variation that makes it easier for the models to learn. Shifting our attention to the CLIP-prefix models, a notable observation is the higher BLEU-1 and BLEU-2 metrics for the Flickr30k dataset compared to those of the former. This indicates that the generated captions exhibit more matching unigrams and bigrams. However, the ordering and overall quality of these captions are less conforming to the references, upon which the Flickr8k models excel, benefiting from encountering less diverse patterns within their dataset.

Table 1. Result comparison of models trained on Flickr8k.

Models	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE
CLIP-prefix (Original)	0.698	0.508	0.363	0.259	0.257	0.565
Ours: CLIP-prefix – Gradient clipping	**0.700**	**0.513**	**0.372**	**0.262**	**0.257**	**0.569**
Ours: CLIP-prefix – Custom tokenizer	0.682	0.497	0.351	0.246	0.249	0.557
Ours: SBG – One caption	0.499	0.276	0.153	0.087	0.167	0.410
Ours: SBG – Top-2 caption	0.520	0.293	0.161	0.089	0.179	0.420
Ours: SBG – Top-5 caption	**0.546**	**0.319**	**0.186**	**0.108**	**0.192**	**0.444**
Merge-RNN	0.601	0.411	0.272	0.179	0.191	0.439

Table 2. Result comparison of models trained on Flickr30k.

Models	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE
CLIP-prefix (Original)	0.715	0.506	0.351	0.243	0.232	0.529
Ours: CLIP-prefix – Gradient clipping	0.715	0.503	0.349	0.235	**0.233**	0.528
Ours: CLIP-prefix – Custom tokenizer	**0.733**	**0.525**	**0.366**	**0.254**	0.232	**0.536**
Ours: SBG – One caption	0.495	0.261	0.139	0.076	0.154	0.378
Ours: SBG – Top-2 caption	0.510	0.279	0.150	0.082	0.164	0.391
Ours: SBG – Top-5 caption	**0.543**	**0.304**	**0.170**	**0.095**	**0.175**	**0.411**
Merge-RNN	0.596	0.404	0.270	0.181	0.175	0.416

Regarding the modifications we added, in Table 1, it is evident that gradient clipping enhances the model's performance. This improvement can be attributed to the relatively small size of the dataset, which has the risk of overfitting. In such cases, gradient clipping functions as a regularization technique, mitigating overfitting and consequently accounting for the observed performance enhancement. Conversely, with the larger dataset size of Flickr30k, the model inherently possesses better generalization capabilities. Thus, gradient clipping not only fails to bring about improvement but, in fact, marginally decreases. Next is the custom tokenizer; interestingly, the situation here contrasts sharply with gradient clipping. Altering the tokenizer essentially transforms the input tokens, necessitating the language model to undergo extensive training to acclimate to the novel patterns introduced. As depicted in Table 1, our model experiences underfitting for this particular configuration, leading to a decrease in performance. When we shift our focus to Table 2, the larger sample size enables the model to leverage the advantages of a reduced token sequence from the tailored tokenizer, resulting in improvements across most metrics.

Our SBG model's performance is relatively modest compared to all the CLIP-prefix variants. This outcome aligns with expectations, considering the wealth of information we missed out on by converting images into keywords. As we augment the generation count, there is a concurrent improvement in performance. This phenomenon is attributed to the inherently noisy nature of this model's input, which involves predicting a caption from a list of words that may lack similarity or exact matches to the source. Consequently, the model becomes sus-

Table 3. Caption comparison of models

Test Image	Real Caption	Method	Prediction on Flickr8k	Prediction on Flickr30k
	1. a boy smiles in front of a stony wall in a city. 2. a little boy is standing on the street while a man in overalls is working on a stone wall. 3. a young boy runs across the street. 4. a young child is walking on a stone paved street with a metal pole and a man behind him. 5. smiling boy in a white shirt and blue jeans in front of a rock wall with a man in overalls behind him.	CLIP-prefix (Original)	a young boy in a blue shirt is walking on a stone path with people behind him.	a young boy in overalls stands in front of a stone wall.
		Ours: CLIP-prefix – Gradient clipping	a boy in a striped shirt is walking through a stone tunnel.	a young boy in overalls and a blue shirt is walking on the street.
		Ours: CLIP-prefix – Custom tokenizer	a boy in a blue shirt and jeans is standing on a stone wall beside a stone wall.	a little boy in a blue shirt is walking along a street.
		Ours: SBG – One caption	a young boy in a blue shirt is smiling.	a boy with a blue shirt looking at a big blue stone.
		Ours: SBG – Top-2 caption	a smiling young boy walking in a park.	a boy runs around a fountain looking sad.
		Ours: SBG – Top-5 caption	a young boy is smiling at the camera in the street.	a young boy smiling in front of an open fountain.
	1. a brown and white dog is running through the snow. 2. a dog is running in the snow. 3. a dog running through snow. 4. a white and brown dog is running through a snow-covered field. 5. the white and brown dog is running over the surface of the snow.	CLIP-prefix (Original)	a white dog is running through the snow.	a white dog is running through the snow.
		Ours: CLIP-prefix – Gradient clipping	a white dog is running through the snow.	a dog running through the snow.
		Ours: CLIP-prefix – Custom tokenizer	a dog running through the snow.	a white dog is running through the snow.
		Ours: SBG – One caption	a brown dog running through the snow.	white dog running in snow.
		Ours: SBG – Top-2 caption	a dog running in the snow.	two small dogs run through the snow on the grass.
		Ours: SBG – Top-5 caption	a brown dog jumps across the snow.	a brown dog is running through the snowy ground.

ceptible to randomness and off-topic predictions. Therefore, any generation that CLIP scores as the most probable is also likely to exhibit more remarkable similarity to the actual reference. The noisy and mismatched nature of the input sequences explains the result in Table 1. Despite a considerable lag in BLEU metrics, the top-5 version showcases METEOR and ROUGE results that outperform those of the Merge-RNN architecture. This suggests that although the model might struggle with using exact words or precisely matching n-grams, it has successfully learned the correct patterns by incorporating alternative, semantically similar words.

4 Conclusion

In summary, we have achieved our primary objective of enhancing the CLIP-prefix architecture, although we acknowledge it as a work in progress.

Considering our use of OpenCLIP, which provides diverse pretrained configurations, there is ample opportunity for additional experiments on different versions of CLIP.

Additionally, numerous variables can be adjusted, such as the content of the hierarchical dictionary. The selection of keys was based on what we deemed sufficient, not on any scientific criteria.

The biggest hurdle we encountered stems from the incongruence between CLIP and the text classifier responsible for constructing the hierarchical dictionary. Given that they were trained on disparate datasets, what they perceive as the most probable elements within an image may not align with one another; any wrong classification by CLIP will impede the proper dictionary traversal. For future work, we plan on addressing these challenges with an attempt at retraining CLIP and more thorough research on the optimal classification for all.

References

1. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
2. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3128–3137 (2015)
3. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: lessons learned from the 2015 MSCOCO image captioning challenge. IEEE Trans. Pattern Anal. Mach. Intell. **39**(4), 652–663 (2017). https://doi.org/10.1109/TPAMI.2016.2587640
4. Tanti, M., Gatt, A., Camilleri, K.: What is the role of recurrent neural networks (RNNs) in an image caption generator? In: Proceedings of the 10th International Conference on Natural Language Generation, pp. 51–60. Association for Computational Linguistics, September 2017. https://aclanthology.org/W17-3506
5. Anderson, P., et al.: Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6077–6086 (2018)

6. Xu, K.: Show, attend and tell: neural image caption generation with visual attention. In: International Conference on Machine Learning. PMLR, pp. 2048–2057 (2015)
7. Chen, L., et al.: SCA-CNN: spatial and channel-wise attention in convolutional networks for image captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5659–5667 (2017)
8. Hoang, D.-H., Dang, D.N.M., Dang-Ngoc, H., Tran, A.-K., Tran, P.-N., Nguyen, C.T.: RBBA: ResNet - BERT - Bahdanau attention for image caption generator. In: The 14th International Conference on ICT Convergence (ICTC 2023), pp. 186–193 (2023)
9. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
10. Goodfellow, I., et al.: Generative adversarial networks. Commun. ACM **63**(11), 139–144 (2020)
11. Chen, C., Mu, S., Xiao, W., Ye, Z., Wu, L., Ju, Q.: Improving image captioning with conditional generative adversarial nets. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 8142–8150, July 2019. https://doi.org/10.1609/aaai.v33i01.33018142
12. Dai, B., Fidler, S., Urtasun, R., Lin, D.: Towards diverse and natural image descriptions via a conditional GAN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2970–2979 (2017)
13. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7008–7024 (2017)
14. Feng, Y., Ma, L., Liu, W., Luo, J.: Unsupervised image captioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4125–4134 (2019)
15. Laina, I., Rupprecht, C., Navab, N.: Towards unsupervised image captioning with shared multimodal embeddings. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7414–7424 (2019)
16. Mokady, R., Hertz, A., Bermano, A.H.: ClipCap: clip prefix for image captioning (2021)
17. Yu, J., Li, H., Hao, Y., Zhu, B., Xu, T., He, X.: CgT-GAN: clip-guided text GAN for image captioning. In: Proceedings of the 31st ACM International Conference on Multimedia. MM 2023. ACM, October 2023. https://doi.org/10.1145/3581783.3611891
18. Radford, A., et al.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
19. Cherti, M., et al.: Reproducible scaling laws for contrastive language-image learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2818–2829 (2023)
20. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3982–3992 (2019)
21. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, June 2015. https://doi.org/10.1109/CVPR.2015.7298682

22. Yin, W., Hay, J., Roth, D.: Benchmarking zero-shot text classification: datasets, evaluation and entailment approach. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics, pp. 3914–3923, November 2019. https://aclanthology.org/D19-1404
23. Hodosh, M., Young, P., Hockenmaier, J.: Framing image description as a ranking task: data, models and evaluation metrics. J. Artif. Intell. Res. **47**, 853–899 (2013). https://api.semanticscholar.org/CorpusID:928608
24. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: new similarity metrics for semantic inference over event descriptions. Trans. Assoc. Comput. Linguist. **2**, 67–78 (2014). https://api.semanticscholar.org/CorpusID:3104920