One Single Hub Text Breaks CLIP: Identifying Vulnerabilities in Cross-Modal Encoders via Hubness

Hiroyuki Deguchi[†] **Katsuki Chousa**[†] **Yusuke Sakai**[‡]
[†]NTT, Inc. [‡]Nara Institute of Science and Technology
{hiroyuki.deguchi,katsuki.chousa}@ntt.com sakai.yusuke.sr9@is.naist.jp

Abstract

The hubness problem, in which hub embeddings are close to many unrelated examples, occurs often in high-dimensional embedding spaces and may pose a practical threat for purposes such as information retrieval and automatic evaluation metrics. In particular, since cross-modal similarity between text and images cannot be calculated by direct comparisons, such as string matching, cross-modal encoders that project different modalities into a shared space are helpful for various cross-modal applications, and thus, the existence of hubs may pose practical threats. To reveal the vulnerabilities of cross-modal encoders, we propose a method for identifying the hub embedding and its corresponding hub text. Experiments on image captioning evaluation in MSCOCO and nocaps along with image-to-text retrieval tasks in MSCOCO and Flickr30k showed that our method can identify a single hub text that unreasonably achieves comparable or higher similarity scores than human-written reference captions in many images, thereby revealing the vulnerabilities in cross-modal encoders.

1 Introduction

Cross-modal encoders such as CLIP (Radford et al., 2021; Schuhmann et al., 2022; Fang et al., 2024; Chen et al., 2023), which can calculate the semantic similarity between texts and images, are widely utilized for various applications, e.g., automatic evaluation metrics for caption quality and retrievers in cross-modal information retrieval, and have become fundamental technologies for multimodal processing. Nevertheless, these models suffer from reliability issues due to the hubness problem (Radovanović et al., 2010), in which a single example exhibits unreasonably high similarity scores with many irrelevant examples. While several countermeasures for hubness in cross-modal encoders have been proposed (Dinu et al., 2015; Lazaridou et al., 2015; Huang et al., 2019; Wang et al., 2023; Chowdhury et al., 2024), concrete texts that are projected to hub embeddings have not yet been found. In embedding-based automatic evaluation metrics for machine translation, Deguchi et al. (2026) proposed a method for identifying a hub translation that is consistently evaluated as high quality across different inputs and references. While this problem of single-modal encoders can be partially mitigated by combining string matching or other sanity checks, cross-modal similarity between text and images cannot be computed through direct comparisons and instead relies heavily on embeddings; thus, the existence of hubs can deteriorate the reliability of cross-modal encoders.

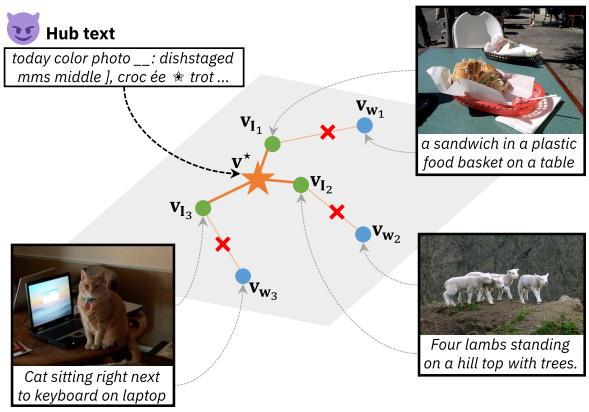


Figure 1: Hub text in cross-modal embedding space. Hub text has higher similarity with many unrelated images than human-written reference captions.

To reveal the vulnerabilities of cross-modal encoders, we propose a method for identifying the hub text, which is a single text that exhibits unreasonably high similarity with many unrelated images. Figure 1 illustrates the hubness problem and an example of hub text identified by our method in the embedding space of a cross-modal encoder. Despite being semantically dissimilar, the hub text is embedded close to many unrelated images in the shared embedding space. We first acquire an optimal hub embedding over a tuning dataset in

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 47242–47256

the continuous embedding space. Since cosine similarity, inner product, and squared Euclidean distance are commonly used to measure similarity in cross-modal tasks, we derive analytical solutions for an optimal hub embedding under these similarity measures. We then decode the hub embedding using an inversion model that reconstructs input texts from their corresponding embeddings (Morris et al., 2023). After decoding, we apply our beam local search to maximize similarity between the hub text and multiple images. We iteratively replace each token in the decoded text with the token that maximizes the average similarity score across the tuning set, while considering multiple candidates. Since our method operates as a black box, i.e., it works with only inputs and their corresponding embeddings, it can be applied to various models.

From our experiments, we confirmed that the proposed method can successfully identify the hub text for various cross-modal encoder models. Specifically, the hub text identified with our method achieved a comparable or higher CLIPSCORE (Hessel et al., 2021) than human reference captions and the hub text identified with the previous method in MSCOCO (Lin et al., 2014; Karpathy and Fei-Fei, 2015) and nocaps (Agrawal et al., 2019). Moreover, we also observed that the contamination of hub text causes a significant drop in accuracy for image-to-text retrieval tasks in MSCOCO and Flickr30k (Young et al., 2014). These findings reveal previously unexplored vulnerabilities in cross-modal encoders that may pose practical threats across a wide range of cross-modal applications, such as the evaluation metrics of CLIPSCORE and cross-modal retrievers.

2 Background and Related Work

Cross-modal encoder Cross-modal encoders, such as CLIP (Radford et al., 2021; Schuhmann et al., 2022; Fang et al., 2024; Chen et al., 2023), project texts and images into a shared embedding space, which enables calculating the similarity between them across modalities. Their models are helpful and widely used for image-to-text retrieval and evaluation metrics of image captioning tasks, e.g., CLIPSCORE (Hessel et al., 2021). We focus on models that have the same architecture as CLIP.

Let $\mathbf{w} \in \mathcal{V}^*$ and $\mathbf{I} \in \mathcal{I}$ be a text and an image, respectively, where $\mathcal{V}^*$ is a Kleene closure of the vocabulary $\mathcal{V}$, and $\mathcal{I} \subseteq \mathbb{R}^{H \times W \times C}$ is a space of normalized images defined by a height $H \in \mathbb{N}$, width $W \in \mathbb{N}$, and number of channels $C \in \mathbb{N}$. The cross-modal encoders $f_\theta \colon \mathcal{V}^* \cup \mathcal{I} \to \mathbb{R}^D$ project texts and images into their corresponding D ($\in \mathbb{N}$)-dimensional embeddings with learned parameters θ. By projecting them into the same space, we can calculate similarity across modalities.

CLIPScore Hessel et al. (2021) proposed CLIPSCORE, an evaluation metric for image captioning tasks. CLIPSCORE evaluates the quality of a caption text $\mathbf{w}$ using the scaled cosine similarity between the caption text and its corresponding image $\mathbf{I}$. The evaluation score is calculated by the similarity function $s \colon \mathbb{R}^D \times \mathbb{R}^D \to [0, M]$ with a scaling factor $M \in \mathbb{R}$, as follows:

$$s(\mathbf{v_w}, \mathbf{v_I}) \coloneqq M \cdot \max\left(\frac{\mathbf{v_w}^\top \mathbf{v_I}}{\|\mathbf{v_w}\|\|\mathbf{v_I}\|}, 0\right), \quad (1)$$

where $\mathbf{v}_\bullet \coloneqq f_\theta(\bullet)$. In general, $M = 2.5$ is used. The corpus-level CLIPSCORE is calculated by averaging scores for all caption–image pairs.

Hubness problem The hubness problem is a phenomenon in which hub embeddings in high-dimensional embedding spaces that are close to many other examples even though they are irrelevant (Radovanović et al., 2010). It causes unexpected behavior in tasks using embedding-based similarity, such as information retrieval and evaluation of text generation tasks, and reduces the reliability of the model. Particularly in cross-modal tasks where direct comparisons of different modalities are impossible, e.g., string matching between two texts, reliance on embeddings is crucial, and several countermeasures for the hubness problem to mitigate its effects have been proposed (Dinu et al., 2015; Lazaridou et al., 2015; Huang et al., 2019; Wang et al., 2023; Chowdhury et al., 2024). Even so, the underlying nature of hub embeddings has not been well studied. Specifically, it remains unclear which specific examples are mapped to the hub embeddings, and whether such hubs are inherent to specific models or arise more generally across architectures and data distributions.

Some studies have identified such vulnerabilities in embedding models. Zhang et al. (2025b) generated adversarial images and audio that received high similarity scores with irrelevant examples in multimodal models. Since images and audio are continuous representations, they can be easily obtained via gradient descent.

Data poisoning Reliable and safe data play a crucial role not only in model training data but also in retrieval-augmented generation (Lewis et al., 2020; Karpukhin et al., 2020; Guu et al., 2020; Ma et al., 2023; Ram et al., 2023; Jeong et al., 2024; An et al., 2025a). One vulnerability of recent large language models (LLMs) is their susceptibility to prompt-based attacks via data poisoning. By injecting malicious or misleading examples into the retrieval data, attackers can manipulate the retrieved content and induce harmful or unintended model behaviors. In response to these issues, recent studies have proposed various attack and defense methods, which suggests an increasing need for security in retrieval systems (Hu et al., 2024; Zou et al., 2025; Zhang et al., 2025a; Tan et al., 2025; Jiao et al., 2025).

Hub text identification Hub text identification is a more challenging task than identifying hub images and audio, as they are discrete representations. In the most naïve method, we need to verify whether texts are mapped to hub embeddings for all possible texts, i.e., discrete token sequences, in $\mathcal{V}^*$ space, which is an NP-hard problem. Deguchi et al. (2026) proposed a 3-step approach for identifying the hub text, and revealed the existence of hub texts in COMET, a neural evaluation metric for translation quality.

They first trained a hub embedding that maximizes evaluation scores over the tuning data. Since the scoring function in COMET is a non-linear feed-forward network, they used gradient descent for producing the hub embedding. They then trained a hub decoder that generates concrete texts from their corresponding text embeddings, i.e., the inverse function of the text encoder, and decoded the hub embedding. Finally, they refined the decoded text to maximize the score by using a greedy local search algorithm. The algorithm sequentially finds the best token that maximizes the score over the tuning data for each token position, which is based on the greedy search.

3 Proposed Method

To reveal vulnerabilities in cross-modal encoders and CLIPSCORE, we propose a method for identifying hub text for their models. We aim to identify a hub text that is close to arbitrary images:

$$\mathbf{w}^\star \coloneqq \operatorname*{argmax}_{\mathbf{w}\in\mathcal{V}^*} \sum_{\mathbf{I}\in\mathcal{I}} s(f_\theta(\mathbf{w}) f_\theta(\mathbf{I})). \tag{2}$$

Our method consists of three steps: **(1) hub acquisition** analytically derives an optimal hub embedding in the embedding space, **(2) hub decoding** decodes the obtained hub embedding into its corresponding concrete text, and **(3) beam local search** refines the hub text to maximize the objective function with the proposed efficient algorithm.

(1) Hub acquisition We first acquire a hub embedding $\mathbf{v}^\star \in \mathbb{R}^D$, which is close to arbitrary image embeddings, by maximizing the objective

$$\mathcal{J}(\mathbf{v}; \mathcal{D}_\mathcal{I}) \coloneqq \frac{1}{|\mathcal{D}_\mathcal{I}|} \sum_{\mathbf{I}\in\mathcal{D}_\mathcal{I}} s(\mathbf{v}, \mathbf{v}_\mathbf{I}), \tag{3}$$

where $\mathcal{D}_\mathcal{I} \subset \mathcal{I}$ is a tuning data that consists of multiple images. In contrast to prior work, which relied on gradient descent to obtain the hub embedding because the COMET scoring function is a non-linear feed-forward network, we derive an analytical solution utilizing the nature of the scoring function s in CLIPSCORE. The optimal solution can be obtained simply by averaging all normalized image embeddings over the tuning data, as follows:

$$\mathbf{v}^\star \coloneqq \operatorname*{argmax}_{\mathbf{v}\in\mathbb{R}^D} \mathcal{J}(\mathbf{v}; \mathcal{D}_\mathcal{I}) = \frac{1}{|\mathcal{D}_\mathcal{I}|} \sum_{\mathbf{I}\in\mathcal{D}_\mathcal{I}} \frac{\mathbf{v}_\mathbf{I}}{\|\mathbf{v}_\mathbf{I}\|}. \tag{4}$$

This solution is derived from the equality condition of the Cauchy–Schwarz inequality. We further provide the detailed derivation of an optimal hub embedding on not only cosine similarity but also other widely used similarity functions, inner product, and squared Euclidean distance, in Appendix C.

(2) Hub decoding Next, we decode the hub embedding into its corresponding concrete text. Since most cross-modal encoders f are non-linear functions and their inverse functions f^{-1} cannot be calculated exactly, we train an inversion model ϕ that reconstructs input texts from their embeddings (Morris et al., 2023). It is trained by minimizing the following negative log-likelihood loss:

$$\mathcal{L}(\phi; \mathcal{D}_{\mathcal{V}^*}) \coloneqq -\sum_{\mathbf{w}\in\mathcal{D}_{\mathcal{V}^*}} \log p_\phi(\mathbf{w}|f_\theta(\mathbf{w})), \tag{5}$$

$$\hat{\phi} \coloneqq \operatorname*{argmin}_\phi \mathcal{L}(\phi; \mathcal{D}_{\mathcal{V}^*}), \tag{6}$$

where $\mathcal{D}_{\mathcal{V}^*} \subset \mathcal{V}^*$ is a training data for the inversion model. During decoding, we generate multiple hypotheses of the hub text $\mathcal{H} \subset \mathcal{V}^*$ from the hub embedding $\mathbf{v}^\star$, obtained in Step (1), using the trained inversion model $\hat{\phi}$:

$$\mathcal{H} \coloneqq \{\mathbf{w}_i\}_{i=1}^{|\mathcal{H}|}, \quad \mathbf{w}_i \sim p_{\hat{\phi}}(\mathbf{w}|\mathbf{v}^\star). \tag{7}$$

Algorithm 1: Beam local search

Given : Scoring function $s\colon \mathbb{R}^D \times \mathbb{R}^D \to [0, M]$ and function Top-$k\colon 2^{\mathcal{V}^* \times [0,M]} \to \{\mathcal{B} \mid \mathcal{B} \subseteq 2^{\mathcal{V}^* \times [0,M]} \wedge |\mathcal{B}| = k\}$ that returns the top-k ($\in \mathbb{N}$) candidates.
Input : Initial hub text candidate $\mathbf{w}^{\text{init}} \in \mathcal{V}^*$ and tuning data $\mathcal{D}_{\mathcal{I}} \subset \mathcal{I}$.
Output : Hub text $\mathbf{w}^\star \in \mathcal{V}^*$.

1 $t \leftarrow 0$, $\mathcal{B}^{(0)} \leftarrow \{(\mathbf{w}^{\text{init}}, \mathcal{J}(f_\theta(\mathbf{w}^{\text{init}}); \mathcal{D}_{\mathcal{I}}))\}$
2 Initialize a new HashMap $\boldsymbol{\mathcal{P}}\colon \mathcal{V}^* \to 2^{\mathbb{N}}$
3 $\boldsymbol{\mathcal{P}}(\mathbf{w}^{\text{init}}) \leftarrow \{1, \ldots, |\mathbf{w}^{\text{init}}|\}$
4 **repeat**
5 $t \leftarrow t + 1$
6 $\mathcal{C} \leftarrow \mathcal{B}^{(t-1)}$
7 **for each** $(\mathbf{w}, S) \in \mathcal{B}^{(t-1)}$ **do**
8 **if** $\boldsymbol{\mathcal{P}}(\mathbf{w}) = \varnothing$ **then**
9 **continue**
10 $i \sim \boldsymbol{\mathcal{P}}(\mathbf{w})$
11 **for each** $v \in \mathcal{V}$ **do**
 `// ∘ denotes concatenation.`
 `// w_{a:b} denotes the slice of w from a to b inclusive.`
12 $\mathbf{w}^{\text{cand}} \leftarrow \mathbf{w}_{1:i-1} \circ v \circ \mathbf{w}_{i+1:|\mathbf{w}|}$
13 $S^{\text{cand}} \leftarrow \mathcal{J}(f_\theta(\mathbf{w}^{\text{cand}}); \mathcal{D}_{\mathcal{I}})$
14 $\mathcal{C} \leftarrow \mathcal{C} \cup \{(\mathbf{w}^{\text{cand}}, S^{\text{cand}})\}$
15 $\boldsymbol{\mathcal{P}}(\mathbf{w}) \leftarrow \boldsymbol{\mathcal{P}}(\mathbf{w}) \setminus \{i\}$
16 $\mathcal{B}^{(t)} \leftarrow$ Top-$k(\mathcal{C})$
17 **if** $\mathcal{B}^{(t)} \neq \mathcal{B}^{(t-1)}$ **then**
18 Initialize a new HashMap $\boldsymbol{\mathcal{P}}\colon \mathcal{V}^* \to 2^{\mathbb{N}}$
19 **for each** $(\mathbf{w}, S) \in \mathcal{B}^{(t)}$ **do**
20 $\boldsymbol{\mathcal{P}}(\mathbf{w}) \leftarrow \{1, \ldots, |\mathbf{w}|\}$
21 **until** $\forall (\mathbf{w}, S) \in \mathcal{B}^{(t)}.\boldsymbol{\mathcal{P}}(\mathbf{w}) = \varnothing$
22 **return** $\underset{\mathbf{w}:(\mathbf{w},S)\in\mathcal{B}^{(t)}}{\operatorname{argmax}} S$

Then, we select the best hypothesis that maximizes the objective function $\mathcal{J}$ over the tuning data $\mathcal{D}_{\mathcal{I}}$:

$$\underset{\mathbf{w}\in\mathcal{H}}{\operatorname{argmax}}\, \mathcal{J}(f_\theta(\mathbf{w}); \mathcal{D}_{\mathcal{I}}). \tag{8}$$

(3) Beam local search Finally, we refine the decoded hub text to maximize the score. Our search algorithm (Algorithm 1) receives the initial hub text, which is the decoded hub text obtained by Step (2), and the main loop iteratively replaces a token in each candidate sequence with a token that maximizes the corpus-level score. In Line 2, we introduce a hash map $\boldsymbol{\mathcal{P}}$ to manage search states and check for convergence. The algorithm terminates when no token updates occur for any candidate in the beam, as specified by the stopping criterion in Line 21. This design enables a more flexible search. While prior work sequentially replaces each token in the decoded text in a left-to-right manner, our method allows tokens to be replaced in a random order (Line 10). The most significant improvement over the conventional method is the extension from greedy search to beam search, which maintains multiple candidates and enables a more efficient search of a larger space.

4 Experiments

4.1 Hub text identification

We identified and evaluated hub texts of CLIP (Radford et al., 2021), LION-CLIP (Schuhmann et al., 2022), DFN-CLIP (Fang et al., 2024), and Alt-CLIP (Chen et al., 2023) in image captioning and image–text retrieval tasks. For all models, we used the validation set of the MSCOCO dataset (Lin et al., 2014; Karpathy and Fei-Fei, 2015) for the tuning data $\mathcal{D}_{\mathcal{I}}$. In all analyses, if no model name is indicated, we used `openai/clip-vit-base-patch32`. We compared the single hub text identified by our method (Ours) with that identified by the sequential greedy local search (GLS) proposed by Deguchi et al. (2026).

Inversion model training We trained inversion models for each cross-modal encoder from mT5-base[1] (Xue et al., 2021), a pretrained encoder-decoder model, with frozen text embeddings encoded by each cross-modal encoder. For the training data $\mathcal{D}_{\mathcal{V}^*}$, we used the caption texts in the MSCOCO training set. The inversion models were optimized using the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$) (Loshchilov and Hutter, 2019) with a learning rate of 3×10^{-4}, 4,000 warm-up steps, and training for 20 epochs using a batch size of 128 captions.

Hub decoding During decoding, we generated 4,096 caption hypotheses from a single hub embedding obtained by Step (1) and then selected the best one using the tuning data. To diversify the hypotheses, we used epsilon sampling with $\varepsilon = 0.02$ (Hewitt et al., 2022; Freitag et al., 2023). Note that this step does not require expensive computational resources because the decoding cost is equivalent to generating 4,096 sentences with mT5-base, which takes less than 1 minute on a single GPU.

Beam local search We applied beam local search to the decoded texts with multiple beam sizes $k \in \{5, 10, 20\}$ and selected the best result on the tuning data. For `openai/clip-vit-base-patch32`, the beam local search replaced tokens 382 times with a sequence length of 23 tokens. It took 12,486 seconds on 8 NVIDIA RTX™ 6000Ada GPUs.

[1] `google/mt5-base`

	MSCOCO (in-domain)				nocaps (out-of-domain)			
	Caption text		Hub text		Caption text		Hub text	
Model	BLIP-2	Human	GLS	Ours	BLIP-2	Human	GLS	Ours
`openai/clip-vit-base-patch32`	0.739	0.759	0.732	†**0.842**	0.740	0.758	0.700	†**0.814**
`openai/clip-vit-large-patch14`	0.613	0.639	0.633	†**0.649**	0.608	**0.623**	0.612	†0.622
`openai/clip-vit-large-patch14-336`	0.628	0.654	0.677	†**0.701**	0.616	0.631	0.620	†**0.663**
`laion/CLIP-ViT-L-14-laion2B-s32B-b82K`	0.720	0.748	0.690	†**0.782**	0.719	**0.740**	0.592	†0.729
`laion/CLIP-ViT-H-14-laion2B-s32B-b79K`	0.729	0.771	0.780	†**0.825**	0.728	**0.759**	0.679	†0.716
`laion/CLIP-ViT-g-14-laion2B-s12B-b42K`	0.688	0.721	**0.767**	0.747	0.689	**0.714**	0.666	†0.680
`apple/DFN2B-CLIP-ViT-L-14`	0.678	0.722	0.723	†**0.814**	0.685	0.712	0.646	†**0.737**
`apple/DFN5B-CLIP-ViT-H-14`	0.789	0.838	0.965	†**0.974**	0.809	0.837	0.831	†**0.841**
`apple/DFN5B-CLIP-ViT-H-14-378`	0.777	0.837	0.995	†**1.023**	0.814	**0.841**	0.798	†0.814
`BAAI/AltCLIP`	0.622	0.635	0.512	†**0.643**	0.622	0.623	0.479	†**0.628**

Table 1: CLIPSCORE evaluated by various models. "Human" indicates reference captions. To evaluate corpus-level scores, each hub text is repeated to match the size of test set, as it is single text. Best and second-best scores for each dataset are indicated in **bold font** and underline, respectively. "†" denotes scores where our identified hub text statistically outperformed "GLS" ($p < 0.05$).

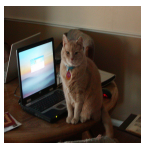
Caption	CLIPSCORE	Text	Image
BLIP-2	0.750	two boys are skateboarding down a street with their skateboards	
Human	0.793	A couple of young boys with skateboards pass a city bus	
Hub	1.012	today color photo __: dishstaged mms middle], croc ée ✯ trot maker gely bw 8 boarded<U+FE0F>: garethapproached cision	
BLIP-2	0.652	a cat sitting on a table	
Human	0.780	Cat sitting right next to keyboard on laptop	
Hub	0.981	today color photo __: dishstaged mms middle], croc ée ✯ trot maker gely bw 8 boarded<U+FE0F>: garethapproached cision	
BLIP-2	0.797	a small kitchen with a stove and microwave oven	
Human	0.806	a kitchen with a small refrigerator and a microwave oven	
Hub	1.034	today color photo __: dishstaged mms middle], croc ée ✯ trot maker gely bw 8 boarded<U+FE0F>: garethapproached cision	

Table 2: Example scores of captions and the single hub text in `openai/clip-vit-base-patch32` on MSCOCO

4.2 Evaluation in image captioning

Setup Our aim here is to investigate whether a hub text is evaluated with an unreasonably high score in image captioning tasks. To validate the effectiveness of our method, we compared the single hub text and caption texts for each image. Specifically, we evaluated for caption texts generated by BLIP-2-FlanT5-XL[2] (Li et al., 2023; Chung et al., 2024), reference captions created by humans, and a single hub text generated by GLS (Deguchi et al., 2026) and by the proposed method. For corpus-level evaluation, we repeated the hub text to match the size of the test set, as it is a single text. We used the test set of the MSCOCO and the validation set[3] of the nocaps dataset (Agrawal et al., 2019) for evaluation. We compared our method with GLS using statistical significance tests via paired bootstrap resampling with 1,000 resamples (Koehn, 2004).

Results Table 1 lists the CLIPSCORE calculated by various cross-modal encoder models. Despite evaluating a single hub text, which is semantically irrelevant to the input images, our method achieved higher scores than reference captions created by humans, image-by-image, on many models. In addition, our method achieved higher scores than the previous method, GLS, in most models. Especially in nocaps, i.e., the out-of-domain dataset, the

[2] `Salesforce/blip2-flan-t5-xl`

[3] The reference captions are available only in the validation set.

Model	Hub text
`openai/clip-vit-base-patch32`	today color photo __: dishstaged mms middle], croc ée ★ trot maker gely bw 8 boarded<U+FE0F>: garethapproached cision
`openai/clip-vit-large-patch14`	photo taken using dnskarchivesdgs unparalleled
`openai/clip-vit-large-patch14-336`	degrees photographer , " toc more " av benefchu (- tely his latest ' buenas wiscondged kirby pa (@

Table 3: Examples of hub texts

	MSCOCO										Flickr30k									
	NDCG		MAP		Recall		Precision		MRR		NDCG		MAP		Recall		Precision		MRR	
#CT	@1	@10	@1	@10	@1	@1k	@1	@5	@1	@10	@1	@10	@1	@10	@1	@1k	@1	@5	@1	@10
`openai/clip-vit-base-patch32`																				
0	50.5	44.3	10.1	33.0	10.1	99.0	50.5	34.2	50.5	60.9	78.0	71.6	15.6	60.2	15.6	99.8	78.0	58.2	78.1	85.4
1	44.2	42.8	8.8	31.4	8.8	99.0	44.2	33.3	44.2	57.2	70.0	68.8	14.0	56.7	14.0	99.8	70.0	55.8	70.0	80.5
1,000	44.1	35.1	8.8	26.3	8.8	52.2	44.1	27.5	44.1	51.5	70.1	56.0	14.0	46.6	14.0	58.6	70.1	46.1	70.1	74.4
`openai/clip-vit-large-patch14-336`																				
0	57.0	49.7	11.4	38.2	11.4	99.3	57.0	38.9	57.0	66.8	87.6	79.8	17.5	69.9	17.5	99.9	87.6	67.0	87.6	92.0
1	52.4	49.0	10.5	37.1	10.5	99.3	52.4	38.3	52.4	64.2	76.5	76.1	15.3	65.0	15.3	99.9	76.5	63.3	76.5	86.1
1,000	52.4	40.7	10.5	31.1	10.5	52.9	52.4	32.0	52.4	59.3	76.5	60.7	15.3	51.8	15.3	60.7	76.5	51.1	76.5	79.5
`laion/CLIP-ViT-g-14-laion2B-s12B-b42K`																				
0	64.7	57.9	13.0	46.5	13.0	99.6	64.7	46.3	64.7	73.5	91.4	85.4	18.3	77.5	18.3	99.9	91.4	73.6	91.4	94.8
1	59.7	55.9	11.9	44.2	11.9	99.7	59.7	44.6	59.7	70.4	88.6	83.9	17.7	75.3	17.7	99.9	88.6	71.4	88.6	93.3
1,000	59.8	47.5	12.0	37.9	12.0	54.5	59.8	38.6	59.8	66.2	88.6	76.4	17.7	68.3	17.7	76.3	88.6	66.9	88.6	91.6
`apple/DFN5B-CLIP-ViT-H-14-378`																				
0	70.4	63.1	14.1	51.9	14.1	99.7	70.4	51.0	70.4	78.4	92.2	87.9	18.4	81.0	18.4	100.0	92.2	77.2	92.2	95.2
1	41.1	55.6	8.2	43.0	8.2	99.8	41.1	46.4	41.1	62.0	63.4	79.1	12.7	68.6	12.7	100.0	63.4	68.5	63.4	80.1
1,000	41.0	26.8	8.2	21.1	8.2	24.2	41.0	21.5	41.0	43.0	63.5	41.8	12.7	34.6	12.7	35.6	63.5	34.6	63.5	64.3
`BAAI/AltCLIP`																				
0	57.9	52.0	11.6	40.6	11.6	99.5	57.9	41.1	57.9	67.8	85.5	80.9	17.1	71.8	17.1	99.9	85.5	68.3	85.5	90.8
1	50.4	49.5	10.1	37.6	10.1	99.5	50.4	38.8	50.4	63.3	69.6	75.8	13.9	64.6	13.9	99.9	69.6	62.4	69.6	82.2
1,000	50.5	38.4	10.1	29.3	10.1	49.3	50.5	30.2	50.5	57.0	69.9	52.4	14.0	43.8	14.0	49.6	69.9	43.7	69.9	72.9

Table 4: Results of image-to-text retrieval tasks with hub text contamination in MSCOCO and Flickr30k. #CT is the number of contaminations, which denotes the number of insertions of the single hub text into the document index.

CLIPSCORE of GLS is lower than that of reference captions, whereas our method outperformed it in five models. Furthermore, we confirmed that our method statistically outperformed GLS in 9 of 10 models on MSCOCO and all models on nocaps, demonstrating that our method identifies hub texts more effectively than GLS.

We present the concrete hub text identified by our method and its CLIPSCORE in Table 2. The hub text is a single string unrelated to any images, yet it achieves unreasonably higher scores than human reference captions. Interestingly, the hub text contains terms such as "color" and "photo", which are presumably likely to appear frequently in the training data of CLIP. Other examples of hub texts are listed in Table 3. The hub texts for other models also contain "photo" or "photographer". These findings suggest that the hub text may occur due to the training data distribution.

	NDCG		MAP		Recall		Precision		MRR	
#CT	@1	@10	@1	@10	@1	@1k	@1	@5	@1	@10
0	50.5	44.3	10.1	33.0	10.1	99.0	50.5	34.2	50.5	60.9
Injected text: Randomly selected caption text										
1	50.5	44.4	10.1	33.0	10.1	99.0	50.5	34.2	50.5	60.9
1,000	50.5	44.4	10.1	33.0	10.1	98.6	50.5	34.2	50.5	60.9
Injected text: Single hub text										
1	44.2	42.8	8.8	31.4	8.8	99.0	44.2	33.3	44.2	57.2
1,000	44.1	35.1	8.8	26.3	8.8	52.2	44.1	27.5	44.1	51.5

Table 5: Comparisons of image-to-text retrieval performance on MSCOCO with `openai/clip-vit-base-patch32` under random caption and hub text injection

4.3 Image-to-text retrieval

Setup We experimented with image-to-text (I2T) retrieval tasks on MSCOCO and Flickr30k (Young et al., 2014) using MTEB (Muennighoff et al., 2023). Assuming attackers contaminate the docu-

ments to be searched, we insert the single hub text for each model into the document side. We also simulated attack cases where the single hub text is repeated and inserted multiple times for search engine optimization (SEO) or cracking. The retrieval performance was evaluated using the normalized discounted cumulative gain (NDCG) at top-1 and top-10, mean average precision (MAP) at top-1 and top-10, recall at top-1 and top-1,000, precision at top-1 and top-5, and mean reciprocal rank (MRR) at top-1 and top-10.

Results The results of the I2T retrieval tasks are demonstrated in Table 4. Here, "#CT" denotes the number of contaminations, which means the number of insertions of the single hub text into the document index. "#CT = 0" is the baseline retrieval performance of each model. We observed that all metrics at top-1 significantly degraded, regardless of the model, even though just a single text was inserted, i.e., when #CT = 1. In particular, precision at top-1 decreased by up to 29.3%, which means that the top-1 search result is often contaminated by the hub text, even though relevant texts are included in the search index. We also evaluated the retrieval performance at #CT = 1,000, which corresponds to a different scenario rather than a single-injection setting. Our intention here is to approximate large-scale duplication of similar content for exploiting search engines and recommendation systems, e.g., template-based generation and spam replication. In this setting, recall at top-1,000 significantly decreased by up to 75.5%. This indicates that contamination using the hub text could hinder the retrieval of relevant texts.

To verify that the observed performance degradation was due to the hub text, we also compared the retrieval performance when a caption randomly selected from the training data was injected. Table 5 shows performance comparisons on MSCOCO for random caption injection and hub text injection. Injecting a random caption did not degrade performance, whereas injecting the hub text did. These results indicate that hub texts, unlike other captions, pose a potential threat.

5 Discussion

5.1 Statistics of instance-level scores

We investigated the statistics of instance-level scores in the captioning evaluation experiments. Table 6 shows the instance-level win rates in CLIPSCORE compared with human reference captions.

	Win rate: Hub > Human			
	MSCOCO		nocaps	
Model	GLS	Ours	GLS	Ours
`openai/clip-vit-base-patch32`	39.1	78.6	27.5	71.1
`openai/clip-vit-large-patch14`	48.4	54.7	45.2	49.5
`openai/clip-vit-large-patch14-336`	60.1	67.7	47.4	62.6
`laion/CLIP-ViT-L-14-laion2B-s32B-b82K`	33.4	60.1	14.2	44.3
`laion/CLIP-ViT-H-14-laion2B-s32B-b79K`	52.8	65.9	26.2	35.4
`laion/CLIP-ViT-g-14-laion2B-s12B-b42K`	63.4	58.2	35.2	38.4
`apple/DFN2B-CLIP-ViT-L-14`	51.4	76.6	29.6	56.8
`apple/DFN5B-CLIP-ViT-H-14`	81.8	83.9	48.0	51.1
`apple/DFN5B-CLIP-ViT-H-14-378`	87.3	90.0	37.5	41.1
`BAAI/AltCLIP`	12.9	52.1	9.0	51.8

Table 6: Instance-level win rates (%) in CLIPSCORE compared with human reference captions

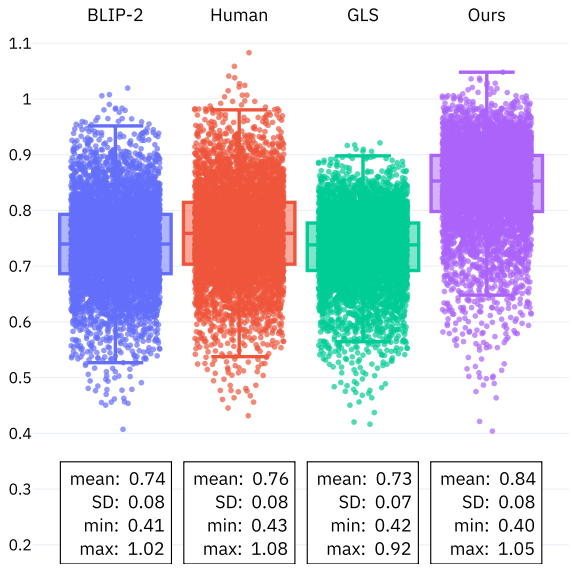


Figure 2: Scatter plots of instance-level scores. Our identified single hub text achieved higher scores than BLIP-2, Human, and GLS in many instances.

The single hub text found with our method achieved higher scores than human reference captions in many models. For both datasets, our proposed method identified more influential hub texts than conventional methods in most models. Specifically, in MSCOCO, our found hub text outperformed reference captions in more than half of the test cases. Furthermore, even models that have high correlations with human assessments (Gomes et al., 2025), `apple/DFN5B-CLIP-ViT-H-14-378`, received higher scores than the reference captions in 90% of test cases, suggesting that hub texts exist regardless of the model performance.

We also compared the instance-level scores between captions and hub texts. Figure 2 shows scatter plots of the instance-level scores in MSCOCO.

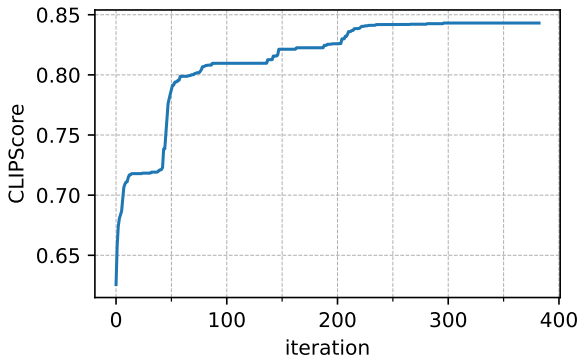


Figure 3: Best CLIPSCORE in beam for each iteration in beam local search

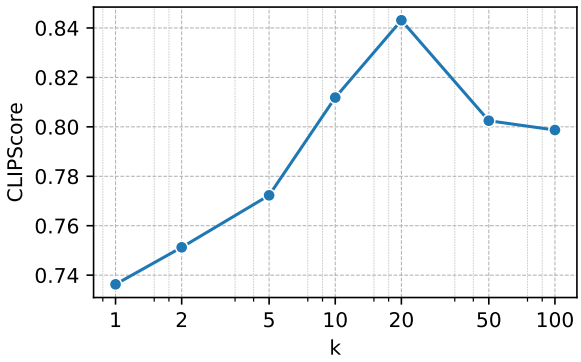


Figure 5: CLIPSCORE when varying beam size k in MSCOCO validation set

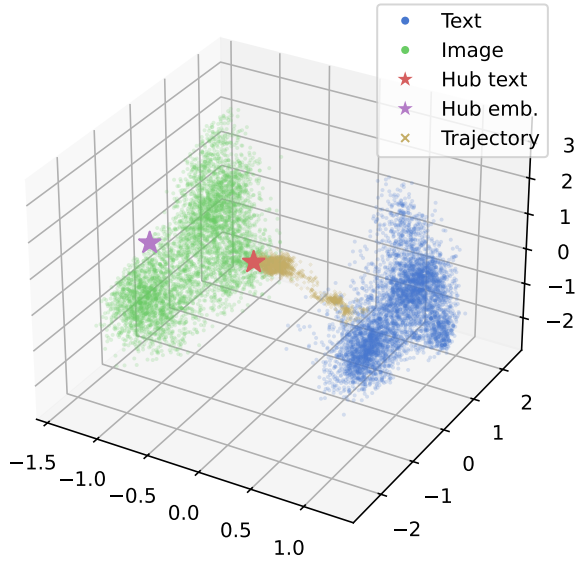


Figure 4: Scatter plots of embeddings in MSCOCO

Human reference captions achieved higher scores than captions generated using BLIP-2 and the single hub text identified with GLS, while the hub text identified with our method significantly outperformed the reference captions.

5.2 Trajectory in local search

To clarify the behavior of our algorithm, we tracked the trajectory of local search. Figure 3 shows the best score in the beam for each iteration of our beam local search in MSCOCO. Because our approach is based on a hill-climbing algorithm, the scores monotonically increased. In early iterations, the score significantly increased, and then the improvement margin became smaller and converged.

Furthermore, we also investigated the trajectory of local search in the embedding space. Figure 4 plots the embeddings of the test cases, optimal hub, hub text finally obtained with our method, and the trajectory in MSCOCO. We used principal component analysis (PCA) to reduce the dimensionality of these embeddings to three dimensions for visualization. As shown here, the hub text left the cluster of text embeddings through local search and moved toward the center of the image cluster. The finally obtained hub text behaves more like an image within the embedding space, even though the input representation is text, which suggests that this results in unreasonably high similarities to images. From these analyses, we found that the hubness problem may be related to the modality gap problem (Liang et al., 2022; An et al., 2025b). Investigating the root cause of the hubness problem and developing methods to mitigate it by leveraging the properties of the embedding space remain important directions for future work.

5.3 Beam size in local search

We varied beam sizes $k \in \{1, 5, 10, 20, 50, 100\}$ in the local search, with the results shown in Figure 5. For $k \leq 20$, the score increases as the beam size is increased. While the score decreases for $k \geq 50$ compared to $k = 20$, it is always better than $k = 1$. This demonstrates that our beam local search is more effective than the conventional greedy local search. In addition, as k increases, computation time grows linearly, and for $k \geq 50$, it became extremely time-consuming. Therefore, we tuned the beam size k from $\{5, 10, 20\}$ for each model.

5.4 Complexity and implementation details

Time complexity of local search Hub text identification is an NP-hard problem because it is determined by computing the scores of all possible texts in $\mathcal{V}^*$. As with the conventional method, the local search is time-consuming. The time complexity of the beam local search is $\mathcal{O}(kT|\mathcal{V}||\mathcal{D}_\mathcal{I}|)$, where

$T \in \mathbb{N}$ is the number of iterations until convergence in the main loop. Since it updates until all tokens are replaced for all beams by checking them with $\mathcal{P}$, the number of iterations T roughly scales linearly with the average sentence length. Our method is slower than the conventional method, due to the addition of a for-loop for the beam search.

Efficient local search using boss–worker pattern To leverage the parallel computing capabilities of multiple GPUs, we implemented the beam local search based on the boss–worker design pattern with a single main process and multiple child processes. The boss submits tasks to a task queue, and each worker associated with the corresponding GPU receives and executes them. Specifically, a task consists of the current state and a query for the token position that will be replaced next. Each worker receives a task and computes the evaluation scores of candidate texts where the i-th token is replaced with the candidate token. In the loop of Line 11 in Algorithm 1, instead of evaluating all candidate texts over the entire vocabulary, each worker is responsible for only a subset of the vocabulary. The boss aggregates the evaluation scores computed by workers and then updates the beams with the Top-k function in Line 16. In addition, to reduce the amount of data in the inter-process communication, we implemented a mechanism where each worker returns only the top-k results, i.e., the boss receives only the $k \times$ #workers $\ll |\mathcal{V}|$ candidates for each beam.

6 Conclusion

This study attempts to find the hub embedding and hub text in cross-modal encoders to identify the vulnerabilities of their models. To this end, we propose methods for analytically finding the optimal hub embedding and finding the hub text more effectively than the prior work, i.e., the beam local search algorithm. Our experiments with image captioning and image-to-text retrieval tasks showed that the hub text found with the proposed method consistently received higher similarity even with unrelated images compared to both the reference captions and the hub text found with the previous method. A key finding is that even powerful models exhibiting high correlations with human assessments are susceptible to hub texts. To develop more reliable models, it is crucial to evaluate not only benchmark scores but also robustness against such malicious attacks.

Limitations

Time complexity The time complexity of beam local search is high, as discussed in Section 5.4. When using larger encoder models or increasing the beam size, it is extremely time-consuming, even if employing our efficient implementation.

Optimization for sequence length in local search The beam local search optimizes only the fixed-length token sequence and does not alter the sequence length. While searching over variable-length sequences could potentially yield hub texts with higher scores, allowing insertion and deletion significantly increases the search space, leading to exponential growth in computational complexity and making it infeasible to complete the search in a realistic time. Thus, we restrict our algorithm to the fixed-length space of the decoded hub text. Developing methods for efficient variable-length search remains a challenge for future work.

Detectability The identified hub text is non-natural, so it could be detected by using other encoders or language models. Our research argues for the necessity of such filtering. While our results make it seem self-evident that such filtering can prevent these issues, we can consider such filtering because our work has revealed the existence of the hub text across various CLIP models. This is supported by the fact that real-world search engines, recommendation systems, and question-answering systems do not always incorporate such mechanisms, which further emphasizes the need for this research to advocate for such filtering.

Furthermore, calculating PPL for all instances is expensive, and over-filtering might degrade performance for the core objective of "finding relevant instances." For example, instances with high PPL due to many technical terms or neologisms may be filtered out, thereby diminishing the usefulness of search engine or recommendation system.

Developing methods to efficiently filter hub texts while maintaining downstream task performance remains an important challenge for future work.

Ethical Considerations

Potential for abuse While our algorithm could potentially be exploited to generate hub texts, the risk of misuse can be reduced through the use of trusted data sources and responsible system design. In practice, recent efforts have been made to improve the security of data pipelines. For example,

the `trust_remote_code` option in HuggingFace datasets (Lhoest et al., 2021), which previously allowed the execution of arbitrary remote code, is no longer supported as of version 4.0.0. This change reduces the risk of injecting malicious or contaminated examples into datasets during download.

Through this work, we provide analyses of vulnerabilities caused by the well-known hubness problem, with the goal of facilitating research on safeguards and mitigation strategies. Since hubness has been observed in a wide range of embedding spaces, we report our findings to raise awareness of its potential implications. Importantly, this study is limited to identifying hub texts that lead to high similarity scores, i.e., it does NOT involve the extraction of personal information or the generation of harmful or malicious content.

We also emphasize the importance of evaluating not only benchmark scores but also safety and robustness against attacks, such as the hub text. We believe that this study will help identify vulnerabilities of cross-modal encoders and contribute to elucidating the causes and nature of hubness.

Co-ordinated disclosure We focused on the vulnerability of cross-modal encoders, falling under the category of coordinated disclosure. While we successfully identified the hub texts in cross-modal encoders, hubness is already widely known as a weakness and vulnerability in embedding spaces (Radovanović et al., 2010), and several methods for mitigating such weakness have been proposed (Dinu et al., 2015; Lazaridou et al., 2015; Huang et al., 2019; Wang et al., 2023; Chowdhury et al., 2024). This fact is further supported by a publicly available paper (Deguchi et al., 2026), which alarms the vulnerability caused by the hubness problem. This paper primarily focused on analyzing the well-known hubness problem in detail, including how hubs appear in the concrete text space and are projected into the embedding space. Thus, our work is not concerned with discovering any new unknown vulnerabilities but rather with investigating the nature of an existing weakness to better understand its mechanisms, aiming to solve the problem at its root. As such, this paper does not fall under the definition of coordinated disclosure as stated in "*e. Works detailing new (i.e. previously not public) security weaknesses/failures without any documentation of co-ordinated disclosure may be in breach of policy.*" of the ACL Policy on Publication Ethics[4].

References

Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. 2019. nocaps: novel object captioning at scale. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8948–8957.

Bang An, Shiyue Zhang, and Mark Dredze. 2025a. RAG LLMs are not safer: A safety analysis of retrieval-augmented generation for large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5444–5474, Albuquerque, New Mexico. Association for Computational Linguistics.

Na Min An, Eunki Kim, James Thorne, and Hyunjung Shim. 2025b. I0T: Embedding standardization method towards zero modality gap. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27182–27199, Vienna, Austria. Association for Computational Linguistics.

Zhongzhi Chen, Guang Liu, Bo-Wen Zhang, Qinghong Yang, and Ledell Wu. 2023. AltCLIP: Altering the language encoder in CLIP for extended language capabilities. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8666–8682, Toronto, Canada. Association for Computational Linguistics.

Neil Chowdhury, Franklin Wang, Sumedh Shenoy, Douwe Kiela, Sarah Schwettmann, and Tristan Thrush. 2024. Nearest neighbor normalization improves multimodal retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22571–22582, Miami, Florida, USA. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Hiroyuki Deguchi, Katsuki Chousa, and Yusuke Sakai. 2026. Hacking neural evaluation metrics with single hub text. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages

[4] https://www.aclweb.org/adminwiki/index.php/ACL_Policy_on_Publication_Ethics#Co-ordinated_disclosure

198–206, Rabat, Morocco. Association for Computational Linguistics.

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. 2015. Improving zero-shot learning by mitigating the hubness problem. *Preprint*, arXiv:1412.6568.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander T Toshev, and Vaishaal Shankar. 2024. Data filtering networks. In *The Twelfth International Conference on Learning Representations*.

Markus Freitag, Behrooz Ghorbani, and Patrick Fernandes. 2023. Epsilon sampling rocks: Investigating sampling strategies for minimum Bayes risk decoding for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9198–9209, Singapore. Association for Computational Linguistics.

Goncalo Emanuel Cavaco Gomes, Chrysoula Zerva, and Bruno Martins. 2025. Evaluation of multilingual image captioning: How far can we get with CLIP models? In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5171–5190, Albuquerque, New Mexico. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Realm: Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

John Hewitt, Christopher Manning, and Percy Liang. 2022. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3414–3427, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhibo Hu, Chen Wang, Yanfeng Shu, Hye-Young Paik, and Liming Zhu. 2024. Prompt perturbation in retrieval-augmented generation based large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 1119–1130, New York, NY, USA. Association for Computing Machinery.

Jiaji Huang, Qiang Qiu, and Kenneth Church. 2019. Hubless nearest neighbor search for bilingual lexicon induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4072–4080, Florence, Italy. Association for Computational Linguistics.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7036–7050, Mexico City, Mexico. Association for Computational Linguistics.

Yang Jiao, Xiaodong Wang, and Kai Yang. 2025. Prattack: Coordinated prompt-rag attacks on retrieval-augmented generation in large language models via bilevel optimization. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 656–667, New York, NY, USA. Association for Computing Machinery.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. 2015. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 270–280, Beijing, China. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, and 13 others. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Zou. 2022. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. In *Advances in Neural Information Processing Systems*.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Yusuke Matsui, Ryota Hinami, and Shin'ichi Satoh. 2018. Reconfigurable inverted index. In *ACM International Conference on Multimedia (ACMMM)*, pages 1715–1723.

John Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander Rush. 2023. Text embeddings reveal (almost) as much as text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12448–12460, Singapore. Association for Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Peter Orlik and Hiroaki Terao. 1992. *Introduction*, pages 1–21. Springer Berlin Heidelberg, Berlin, Heidelberg.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(86):2487–2531.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. 2022. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Xue Tan, Hao Luan, Mingyu Luo, Xiaoyan Sun, Ping Chen, and Jun Dai. 2025. RevPRAG: Revealing poisoning attacks in retrieval-augmented generation through LLM activation analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12999–13011, Suzhou, China. Association for Computational Linguistics.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. 2025. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *Preprint*, arXiv:2502.14786.

Yimu Wang, Xiangru Jian, and Bo Xue. 2023. Balance act: Mitigating hubness in cross-modal retrieval with query and gallery banks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10542–10567, Singapore. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Thomas Zaslavsky. 1975. *Facing up to arrangements : face-count formulas for partitions of space by hyperplanes*. Memoirs of the American Mathematical Society. American Mathematical Society.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11975–11986.

Chenyang Zhang, Xiaoyu Zhang, Jian Lou, Kai Wu, Zilong Wang, and Xiaofeng Chen. 2025a. Poisonedeye: Knowledge poisoning attack on retrieval-augmented generation based large vision-language models. In *Forty-second International Conference on Machine Learning*.

Tingwei Zhang, Fnu Suya, Rishi Jha, Collin Zhang, and Vitaly Shmatikov. 2025b. Adversarial hubness in multi-modal retrieval. *Preprint*, arXiv:2412.14113.

Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. 2025. Poisonedrag: knowledge corruption attacks to retrieval-augmented generation of large language models. In *Proceedings of the 34th USENIX Conference on Security Symposium*, SEC '25, USA. USENIX Association.

A Licenses

Models Table 7 lists the licenses and references of the models we employed in our experiments.

Datasets In the MSCOCO dataset (Lin et al., 2014; Karpathy and Fei-Fei, 2015), the annotations belong to the COCO Consortium and are licensed under a Creative Commons Attribution 4.0 license. The COCO Consortium does not own the copyright of the images, and use of the images must abide by the Flickr Terms of Use. The nocaps dataset (Agrawal et al., 2019) is released under a CC-BY-2.0 License. Flickr30k (Young et al., 2014) is solely provided for non-commercial research and/or educational purposes.

B Intermediate Texts in Local Search

Table 8 shows the trajectory of the intermediate hub texts in beam local search. We only extracted the top-1 candidates for each iteration. Due to space limitations, we included results only for the first three and the last three steps.

C Derivation of Optimal Hub Embedding

C.1 Derivation

Maximize cosine similarity Most cross-modal encoders employ cosine similarity for the scoring function, and the optimal hub embedding can be calculated analytically when cosine similarity is used for the scoring function s. For simplicity, we derive it using pure cosine similarity without any scaling and clipping. From the definition, the optimal hub embedding $\mathbf{v}^\star \in \mathbb{R}^D$ and the objective $\mathcal{J}\colon \mathbb{R}^D \times 2^{\mathbb{R}^D} \to [-1, 1]$ can be formulated as follows:

$$\mathbf{v}^\star \coloneqq \operatorname*{argmax}_{\mathbf{v}\in\mathbb{R}^D} \mathcal{J}(\mathbf{v}; \mathcal{D}_\mathcal{I}), \tag{9}$$

$$\mathcal{J}(\mathbf{v}; \mathcal{D}_\mathcal{I}) \coloneqq \frac{1}{|\mathcal{D}_\mathcal{I}|} \sum_{\mathbf{I}\in\mathcal{D}_\mathcal{I}} \frac{\mathbf{v}^\top \mathbf{v_I}}{\|\mathbf{v}\|\|\mathbf{v_I}\|} \tag{10}$$

$$= \frac{\mathbf{v}^\top}{\|\mathbf{v}\|} \cdot \frac{1}{|\mathcal{D}_\mathcal{I}|} \sum_{\mathbf{I}\in\mathcal{D}_\mathcal{I}} \frac{\mathbf{v_I}}{\|\mathbf{v_I}\|} \tag{11}$$

$$= \frac{\mathbf{v}^\top}{\|\mathbf{v}\|} \bar{\mathbf{v}}, \tag{12}$$

where $\bar{\mathbf{v}} \coloneqq \frac{1}{|\mathcal{D}_\mathcal{I}|} \sum_{\mathbf{I}\in\mathcal{D}_\mathcal{I}} \frac{\mathbf{v_I}}{\|\mathbf{v_I}\|}$, i.e., the averaged vector of all L^2-normalized image embeddings in the tuning data $\mathcal{D}_\mathcal{I}$. Since Equation (12) calculates the inner product of two vectors, the upper bound of the absolute value of the inner product can be obtained using Cauchy–Schwarz inequality:

$$\left| \frac{\mathbf{v}^\top}{\|\mathbf{v}\|} \bar{\mathbf{v}} \right| \leq \left\| \frac{\mathbf{v}}{\|\mathbf{v}\|} \right\| \cdot \|\bar{\mathbf{v}}\| = \|\bar{\mathbf{v}}\|. \tag{13}$$

In Equation (13), the equality condition is when $\mathbf{v}$ and $\bar{\mathbf{v}}$ are parallel. Hence, when $\mathbf{v} \propto \bar{\mathbf{v}} = \frac{1}{|\mathcal{D}_\mathcal{I}|} \sum_{\mathbf{I}\in\mathcal{D}_\mathcal{I}} \frac{\mathbf{v_I}}{\|\mathbf{v_I}\|}$ is satisfied, $\frac{\mathbf{v}^\top}{\|\mathbf{v}\|} \bar{\mathbf{v}} = \mathcal{J}(\mathbf{v}; \mathcal{D}_\mathcal{I})$ is maximized.

Maximize inner product Similar to cosine similarity, the optimal hub embedding in the inner product can also be analytically obtained from image embeddings. In this case, the objective function can be formulated as follows:

$$\mathcal{J}(\mathbf{v}; \mathcal{D}_\mathcal{I}) \coloneqq \frac{1}{|\mathcal{D}_\mathcal{I}|} \sum_{\mathbf{I}\in\mathcal{D}_\mathcal{I}} \mathbf{v}^\top \mathbf{v_I} \tag{14}$$

$$= \mathbf{v}^\top \frac{1}{|\mathcal{D}_\mathcal{I}|} \sum_{\mathbf{I}\in\mathcal{D}_\mathcal{I}} \mathbf{v_I} \tag{15}$$

$$= \mathbf{v}^\top \bar{\mathbf{v}}. \tag{16}$$

The upper bound of the absolute value of the inner product can be obtained, as well as Equation (13):

$$\left| \mathbf{v}^\top \bar{\mathbf{v}} \right| \leq \|\mathbf{v}\| \cdot \|\bar{\mathbf{v}}\| = \|\mathbf{v}\| \cdot \|\bar{\mathbf{v}}\| \cdot \cos 0. \tag{17}$$

Model	License	Reference
`openai/clip-vit-base-patch32`	MIT	(Radford et al., 2021)
`openai/clip-vit-large-patch14`	MIT	(Radford et al., 2021)
`openai/clip-vit-large-patch14-336`	MIT	(Radford et al., 2021)
`laion/CLIP-ViT-L-14-laion2B-s32B-b82K`	MIT	(Schuhmann et al., 2022)
`laion/CLIP-ViT-H-14-laion2B-s32B-b79K`	MIT	(Schuhmann et al., 2022)
`laion/CLIP-ViT-g-14-laion2B-s12B-b42K`	MIT	(Schuhmann et al., 2022)
`apple/DFN2B-CLIP-ViT-L-14`	Apple Machine Learning Research Model	(Fang et al., 2024)
`apple/DFN5B-CLIP-ViT-H-14`	Apple Machine Learning Research Model	(Fang et al., 2024)
`apple/DFN5B-CLIP-ViT-H-14-378`	Apple Machine Learning Research Model	(Fang et al., 2024)
`BAAI/AltCLIP`	CreativeML Open RAIL-M	(Chen et al., 2023)
`google/mt5-base`	Apache 2.0	(Xue et al., 2021)
`Salesforce/blip2-flan-t5-xl`	MIT	Li et al. (2023)

Table 7: Licenses of models we used

a color photo detfrom the 2 0 1 0 s of kingfish 's quash backpack '
a color photo vc from the 2 0 1 0 s evident kingfish 's quash backpack '
a color photo vc from the 2 0 1 0 s evident kingiller 's quash backpack '
⋮
today color photo __: dishstaged mms bays], croc xiii ★ trot knogely bw 7 boarded : garethapproached cision
today color photo __: dishstaged mms middle], croc air ★ trot maker gely bw 7 boarded : garethapproached cision
today color photo __: dishstaged mms middle], croc ée ★ trot maker gely bw 8 boarded : garethapproached cision

Table 8: Intermediate hub texts of `openai/clip-vit-base-patch32` in beam local search

Hence, when $\mathbf{v}$ and $\bar{\mathbf{v}}$ have the same angle, the objective function will be maximized with a large L^2-norm of $\mathbf{v}$.

Minimize squared Euclidean distance In k-nearest neighbor search, the squared Euclidean distance is often used[5]. Since we maximize the objective in this paper, it is defined as follows:

$$\mathcal{J}(\mathbf{v}; \mathcal{D}_\mathcal{I}) \coloneqq -\frac{1}{|\mathcal{D}_\mathcal{I}|} \sum_{\mathbf{I} \in \mathcal{D}_\mathcal{I}} \|\mathbf{v} - \mathbf{v}_\mathbf{I}\|^2. \tag{18}$$

Since $\|\mathbf{v} - \mathbf{v}_\mathbf{I}\|^2 = \|\mathbf{v}\|^2 - 2\mathbf{v}^\top \mathbf{v}_\mathbf{I} + \|\mathbf{v}_\mathbf{I}\|^2$ and $\frac{1}{|\mathcal{D}_\mathcal{I}|} \sum_{\mathbf{I} \in \mathcal{D}_\mathcal{I}} \|\mathbf{v}_\mathbf{I}\|^2$ is a constant under the objective function, Equation (18) can be formulated as follows:

$$(18) \propto -\|\mathbf{v}\|^2 + \frac{1}{|\mathcal{D}_\mathcal{I}|} \sum_{\mathbf{I} \in \mathcal{D}_\mathcal{I}} 2\mathbf{v}^\top \mathbf{v}_\mathbf{I} \tag{19}$$

$$= -\|\mathbf{v}\|^2 + 2\mathbf{v}^\top \frac{1}{|\mathcal{D}_\mathcal{I}|} \sum_{\mathbf{I} \in \mathcal{D}_\mathcal{I}} \mathbf{v}_\mathbf{I} \tag{20}$$

$$= -\|\mathbf{v}\|^2 + 2\mathbf{v}^\top \bar{\mathbf{v}}. \tag{21}$$

Since $g\colon \mathbf{v} \mapsto -\|\mathbf{v}\|^2 + 2\mathbf{v}^\top \bar{\mathbf{v}}$ is a convex quadratic function; thus, the global maxima of g can be obtained as follows:

$$\nabla_\mathbf{v} g(\mathbf{v}) = \nabla_\mathbf{v}(-\|\mathbf{v}\|^2 + 2\mathbf{v}^\top \bar{\mathbf{v}}) \tag{22}$$

$$= -2\mathbf{v} + 2\bar{\mathbf{v}} = 0, \tag{23}$$

$$\therefore \mathbf{v} = \bar{\mathbf{v}}. \tag{24}$$

Therefore, the optimal hub embedding is the average vector of image embeddings.

C.2 Optimality gap between CLIPSCORE and cosine similarity

For all experiments, we used the optimal solution in cosine similarity for hub acquisition because we used the same hub text for not only the image captioning evaluation but also the image-to-text retrieval task, which retrieves the k-nearest neighbor texts based on cosine similarity. We discuss the optimality gap between CLIPSCORE and cosine similarity in this section.

The differences between the two are scaling and clipping, as shown in Equation (1). Since the scaling parameter M is constant with respect to the variable being maximized, the optimal solution is preserved. In contrast, non-linear zero clipping yields differences in the exact optimal solutions. The clipping in CLIPSCORE ignores images that have negative cosine similarity scores with a hub embedding, so the CLIPSCORE is calculated with images that are contained in the hemisphere centered at a fixed embedding in the hypersphere.

[5]For example, Faiss (Johnson et al., 2019; Douze et al., 2024) and Rii (Matsui et al., 2018) use the squared Euclidean distance for the default distance function.

Caption or text	MSCOCO	nocaps
BLIP-2	0.354	0.352
Human	0.381	0.380
GLS	0.356	0.335
Ours	**0.406**	**0.400**

Table 9: Evaluation of image captioning tasks using `google/siglip2-base-patch16-256`

	NDCG		MAP		Recall		Precision		MRR	
#CT	@1	@10	@1	@10	@1	@1k	@1	@5	@1	@10
0	69.4	61.5	13.9	50.1	13.9	99.8	69.4	49.5	69.4	77.6
1	54.2	56.7	10.8	44.4	10.8	99.8	54.2	45.8	54.2	68.7
1,000	54.1	37.8	10.8	29.9	10.8	37.7	54.1	30.5	54.1	57.9

Table 10: Image-to-text retrieval performance of `google/siglip2-base-patch16-256` on MSCOCO

Here, the solution we obtained has positive cosine similarity with all images and is the optimal solution on the hemisphere that contains them. Thus, we obtained the global optima on the fixed hemisphere that contains all images. Using other hemispheres means that the CLIPSCORE for some images will be 0, and the optimized embedding under such hemispheres would not be the hub embedding.

Moreover, considering all hemispheres, including ones that allow CLIPSCORE $= 0$ for some images, can be regarded as a central hyperplane arrangement problem (Orlik and Terao, 1992), and we would need to compute $2\sum_{i=0}^{D-1}\binom{|\mathcal{D}_\mathcal{I}|-1}{i}$ times (Zaslavsky, 1975), which is infeasible.

From the above, we obtained the optimal hub embedding based on cosine similarity that works generically across multiple tasks.

D Hub Text Identification Across Other Encoder Types

To investigate the generality of our method, we also identify hub texts in SigLIP (Zhai et al., 2023), where the models are trained with a sigmoid-based loss function that differs from CLIP. We used `google/siglip2-base-patch16-256` (Tschannen et al., 2025) for the experiments. Table 9 and Table 10 show the results of the caption evaluation and image-to-text retrieval tasks, respectively. These results demonstrated that we also successfully identified the hub text in SigLIP-2 trained using a different loss function. We confirmed that robustness against hubness still remains an issue, even with the latest model.

	Deguchi et al. (2026)	Ours
Target metric	COMET	CLIPSCORE
Scoring function	Feed-forward network	Cosine similarity
Hub embedding	Numerical solution	Analytical solution
Local search	Greedy search	Beam search
Search order	Sequential	Random
Method type	White-box	Black-box

Table 11: Comparison of related work (Deguchi et al., 2026) and our work

Dataset	#instances
MSCOCO (Train)	113,287
MSCOCO (Validation)	5,000
MSCOCO (Test)	5,000
nocaps	4,500
Flickr30k	1,000

Table 12: Statistics of datasets we used

E Comparison of Methods for Hub Text Identification

Table 11 lists the differences between the related work (Deguchi et al., 2026) and our work.

F Dataset Statistics

The dataset statistics are listed in Table 12.