Knowledge-Based Systems 329 (2025) 114400

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Augmented decoding method using semantic diverse beam search for language generation model

HyungSun Na [a,1], Hee-Gook Jun [b,1], Jinhyun Ahn [c], Dong-Hyuk Im [d,*]

[a] Dept. of Artificial Intelligence Convergence, Kwangwoon University, Seoul, Republic of Korea
[b] Finda Inc., Seoul, Republic of Korea
[c] Dept. of Management Information Systems, Jeju National University, Jeju, Republic of Korea
[d] School of Information Convergence, Kwangwoon University, Seoul, Republic of Korea

ARTICLE INFO

Keywords:
Image captioning
Semantic diversity
Beam search
Knowledge graph
Decoding algorithm

ABSTRACT

Image captioning, the task of automatically generating natural language descriptions from visual content, has achieved remarkable accuracy in recent years. However, current approaches face a critical limitation in semantic diversity. Most diversity-oriented methods evaluate similarity at the surface lexical level, incorrectly treating lexically different but semantically equivalent phrases (e.g., 'dog runs' vs 'canine sprints') as meaningfully diverse outputs. This superficial approach fails to capture true semantic variation. Consequently, generated captions appear different but convey essentially identical meanings. To address this fundamental limitation, we propose Semantic Diverse Beam Search (SDBS), an augmented decoding algorithm that operates in semantic space rather than surface lexical space. SDBS integrates four key innovations: knowledge graph-based semantic similarity scoring, adaptive thresholding for important word focus, statistics-based stratified top-k sampling, and beam size normalization. Additionally, we introduce an early-stop strategy that significantly reduces computational complexity while maintaining generation quality, making SDBS practically viable for real-world applications. Comprehensive experiments demonstrate that SDBS achieves superior performance on both traditional metrics and modern evaluation approaches (BARTScore++, LLM-based assessment), generating captions with genuine semantic diversity while maintaining high accuracy and computational efficiency.

1. Introduction

Image captioning represents a fundamental multimodal AI task that bridges computer vision and natural language processing by automatically generating textual descriptions from visual information in input images [1–4]. A vision model identifies objects and their spatial information, while the NLP model analyzes inter-object relationships to produce human-readable sentences. Accordingly, image captioning aims to generate captions that are both accurate and diverse [5–7].

Existing caption generation models have primarily focused on improving accuracy [8,9]. Although recent models have achieved remarkable accuracy in generating factual descriptions, they often produce repetitive and formulaic outputs that lack the rich variety of human expression [10]. This limitation significantly affects practical applications that demand creative or stylistically varied content generation, such as educational tools and accessibility systems. However, diversity [11] constitutes an equally critical aspect, encompassing both lexical variation (i.e., surface-form differences) and semantic diversity (i.e., variation in scene graph content). Ideally, generated outputs should not only enumerate all objects within an image but also articulate their interrelationships from multiple perspectives. Nevertheless, an excessive emphasis on accuracy frequently diminishes both lexical and semantic diversity, leading to the production of overly conservative sentences that converge toward safe and generic descriptions. Conversely, emphasizing surface-level diversity without semantic grounding may compromise factual correctness by encouraging less reliable yet more varied outputs highlighting the inherent trade-off between accuracy and diversity in image captioning [6,12]. Addressing this accuracy-diversity dilemma requires novel approaches capable of generating semantically distinct and meaningful perspectives while preserving factual integrity. The key to solving this challenge lies in the decoding process itself. Recent comprehensive analyses have

* Corresponding author.
E-mail addresses: nayosk@kw.ac.kr (H. Na), heegook@finda.co.kr (H.-G. Jun), jha@jejunu.ac.kr (J. Ahn), dhim@kw.ac.kr (D.-H. Im).
[1] The authors contribute equally.

https://doi.org/10.1016/j.knosys.2025.114400
Received 12 January 2023; Received in revised form 25 August 2025; Accepted 1 September 2025
Available online 2 September 2025
0950-7051/© 2025 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

demonstrated that decoding methods play an integral role in transforming language models into practical task solvers, with the choice of decoding strategy having a considerable impact on both the quality and linguistic properties of generated outputs [13].

Current decoding algorithms face fundamental limitations in achieving semantic diversity as they struggle to generate captions with meaningfully distinct scene graph content rather than mere lexical variations. While beam search generates more accurate sentences than greedy search, it produces outputs that are overly general and monotonous compared to manually generated sentences [6]. To address this limitation, diverse beam search was proposed, which divides candidates into groups and processes beam search for each group while computing diversity scores between group outputs [14]. Although this approach generates more diverse sentences than standard beam search, it fundamentally relies on string-based similarity functions that fail to capture genuine semantic relationships.

Fig. 1(a) shows an example of the similarity metric used in a diverse beam search. For two given words, the similarity score was obtained by dividing the length of the longer word by the number of positions where the same character was located between two words. In Fig. 1(a), "COFFEE" and "ESPRESSO" are identical only in the fifth character 'e', and the similarity obtained by dividing 1 by 8 (the length of "ESPRESSO") is 12.5 %. This example shows that semantically similar words were misjudged as different words. This problem arises from string-match-based comparisons rather than semantic comparisons.

In this study, we propose a semantic similarity-based approach for semantic diversity. Our method measures the semantic differences between words using knowledge graph-based word relationships (Fig. 1(b)). Moreover, we provide additional methods to balance accuracy and diversity. The contributions of this study are as follows.

- We developed a semantic diversity scoring function based on knowledge graph similarity to capture genuine semantic variation rather than superficial lexical differences.
- We introduced an adaptive thresholding mechanism to avoid unnecessary diversity computations for semantically distant word pairs, thereby improving computational efficiency without compromising performance.
- We proposed stratified top-k sampling to focus diversity computations on statistically significant words, maintaining generation quality while reducing computational overhead.
- We applied beam size normalization to decouple beam size from group size, allowing more flexible hyperparameter selection and improving scalability.
- We implemented an early-stop strategy that significantly reduces computational complexity while preserving generation quality, addressing practical concerns in deploying semantically constrained decoding.

The remainder of this study is structured as follows. Section 2 introduces related work. In Section 3, we provide preliminaries and describe the limitations of the existing works. In Section 4, we present our semantic diverse beam search algorithm in detail. Section 5 presents experimental results. Finally, we provide conclusions.

2. Related work

2.1. Neural language generation models

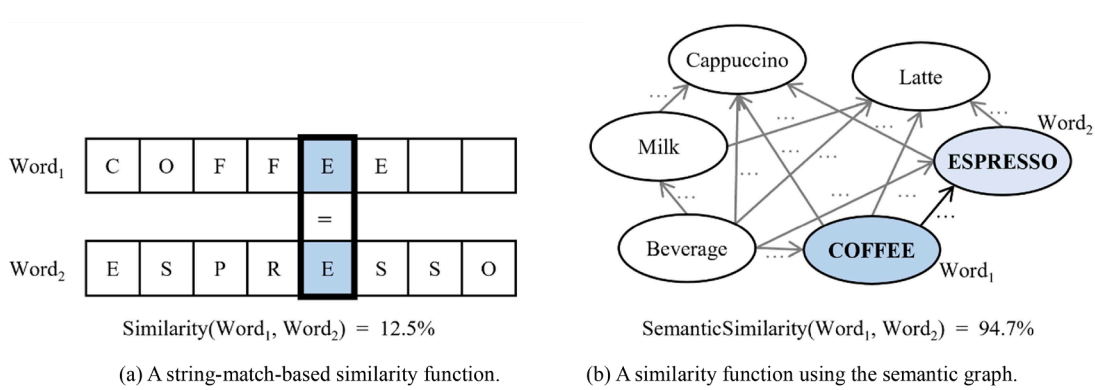
Seq2Seq (Sequence to Sequence) is an encoder-decoder natural language generation model that generates sentences from input sentences. The encoder-decoder-separated structure enables the generation of output sentences of variable lengths without being affected by the input length [15–17].

Seq2Seq, with attention, enables the decoder to refer to all the output hidden states of the encoder at every time step. Attention [18] was presented to improve the Seq2Seq model performance; In Seq2Seq, the last hidden state of the encoder is used only once for an initial input hidden state to the first time step of the decoder. At each time step, the output hidden state of the decoder is used as a query. All output hidden states of the encoder are used as (key, value) pairs that calculate how much they are related to the queries.

The transformer [19] solved a gradient vanishing problem by eliminating sequential models such as recurrent neural network (RNN), Long short-term memory (LSTM), and gated recurrent unit (GRU) from the Seq2Seq model. The transformer built an encoder-decoder model using self-attention layers to replace the sequential models. The model structure without sequential models processes input data simultaneously without going through time steps and improves accuracy by adopting parallel processing techniques.

2.2. Image captioning models

Image captioning is processed in a top-down or bottom-up approach. The top-down approach generates sentences using the entire feature of an image [20–22]. The bottom-up approach generates sentences by extracting small features of objects from an image. Neural image caption (NIC) [23] is a top-down approach that modified the encoder of Seq2Seq by replacing sequential models with CNN. Since then, methods using top-down and bottom-up approaches together have been studied. Methods of adding Attention to the existing top-down model [24,25] and methods of applying a Transformer [26–29] have been studied as the hybrid approach. While these approaches have achieved significant progress in image captioning accuracy, the emergence of large-scale pre-trained models has opened new possibilities for both improving model architectures and developing more sophisticated decoding strategies. Recent developments in large-scale pre-trained models and modern decoding strategies offer new opportunities to enhance both



(a) A string-match-based similarity function. (b) A similarity function using the semantic graph.

Fig. 1. Examples of the similarity computation.

quality and diversity in caption generation.

2.3. Large-Scale pre-trained models and modern decoding

Recent advances in large-scale pre-trained language models have significantly transformed text generation capabilities and decoding strategies. GPT-2 [30] demonstrated remarkable text generation abilities through autoregressive modeling, while GPT-3 [31] further showcased the potential of scale in improving generation quality and diversity. The T5 framework [32] introduced a unified approach to various NLP tasks by treating them as text-to-text problems, providing insights into how pre-trained models can be adapted for generation tasks including image captioning. Modern decoding strategies have evolved to address the diversity-accuracy tradeoff that traditional beam search struggles with. Top-k sampling [33] constrains the candidate pool to the k most probable tokens, while nucleus sampling [34] dynamically adjusts the candidate set based on cumulative probability mass. Recent work on contrastive search [35] has also shown promise in balancing quality and diversity through confidence-based decoding. However, these sampling-based methods primarily operate at the surface lexical level and may not capture semantic diversity effectively. Recent comprehensive analyses [13] have highlighted the importance of developing more sophisticated decoding strategies that can balance quality and diversity in neural text generation.

3. Preliminaries and problem description

This section explains the basic concept of a decoding algorithm that generates sentences in image captioning models. We then describe these limitations and challenging issues from the perspective of semantic diversity.

3.1. Model structure

Our research aimed to build a decoding algorithm with better diversity and accuracy for image captioning. The decoding algorithm was used to generate descriptive captions for input images using the image captioning model. We built an image captioning model in which our decoding algorithm was to be studied (Fig. 2).

In Fig. 2(a), ResNet101 was used as an image encoder model to extract the feature information of the input images. In Fig. 2(b), the text decoder models receive the extracted feature data from the encoder and generate sentences. We used two text decoder models (Transformer and LSTM with Attention) to conduct various experiments using our decoding algorithm. Fig. 2(c) shows our decoding algorithm processed on the Transformer and LSTM with Attention, respectively.

3.2. Terminology

Table 1 describes the notations and terminologies used in the decoding algorithms for image captioning.

The vocabulary set V is the total number of words to be managed by an image captioning model, and $|V|$ is the vocabulary size. Input data X is a set of images input to the image captioning model.

A special token ⟨sos⟩ (start of a sentence) is the first input data of the decoder, and ⟨eos⟩ (end of a sentence) indicates the end of decoding. Decoding time step T represents the size of the time steps required to generate sentences in the decoding process. For example, if a decoder outputs a sentence composed of three words with two special tokens (<sos>, word_1, word_2, word_3, and <eos>), then the number of time steps T is five.

The beam size B is the number of output words generated at each step. For example, B is 1 for a greedy search because the greedy search outputs only one word at each step, and if B is 3 for a beam search, then the beam search outputs three words at each step. The group size G is a

Table 1
Notation and terminology.

Notation	Terminology
V	Vocabulary set
|V|	Vocabulary size
$\boldsymbol{X}$	Input data
$< sos >$	Special token (start of sentence)
$< eos >$	Special token (end of sentence)
$\boldsymbol{T}$	Decoding time step
$\boldsymbol{B}$	Beam size
$\boldsymbol{G}$	Group size
λ	Diversity strength
$[n]$	1, 2, 3, …, n
$\boldsymbol{V}_{[G]}$	$V_1, V_2, V_3, \ldots, V_G$

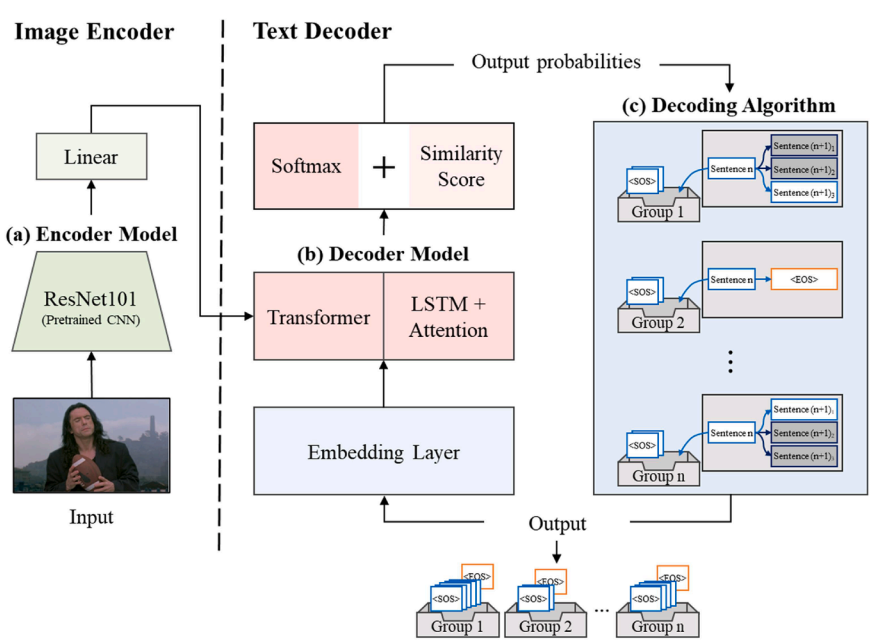


Fig. 2. The image captioning model structure of our study. (a) image encoder model; (b) text decoder models; (c) decoding algorithm.

hyperparameter of a diverse beam search. A diverse beam search divides the input data by G and performs a beam search for each group. Diversity strength λ is a hyperparameter that adjusts the balance between diversity and grammatical soundness.

The symbol $[n]$ represents a numeric sequence from one to n. For example, $V_{[G]}$ represents the number of G sequential vocabulary sets from V_1 to V_g.

3.3. Decoding algorithms

Decoding algorithms are used during the time-stepwise text decoding process. In the first time step, the decoder receives a special token ⟨sos⟩ and predicts a word. After the first step, the decoder receives the previous output word as input data to predict the next word. It repeats the word generation process until the decoder outputs a special token ⟨eos⟩. Decoding algorithms select the best word to output at each time step. Because the decoding cost increases exponentially with the size of the vocabulary set and time step length, it is essential to apply the decoding algorithm to optimize the cost.

In this section, two basic concepts of decoding algorithms–greedy search and beam search–are explained. We then explain the diverse beam search proposed to improve the base algorithms.

3.3.1. Greedy search

A greedy search is a simple and straightforward decoding algorithm (Fig. 3). The decoder receives a special token sos for the first time step. The decoder then uses an output word generated in the previous time step as the input data from the second step.

The softmax layer in the decoder outputs the probabilities of all words in the vocabulary set. In each step, a greedy search selects the word with the highest probability (Eq. (1)).

$$\theta_{(y_t)} = \underset{y}{\operatorname{argmax}}\ logP_r(y_t|y_{[t-1]},\ x), \tag{1}$$

where y_t is the token selected at decoding step t, $y_{[t-1]}$ denotes the sequence of previously generated tokens, and x represents the input context. This formulation builds the output sequence step by step, choosing the most likely next token at each stage based on the model's predicted probabilities.

Eq. (2) describes the score value of the generated sentences using a greedy search, which is the sum of the values computed using Eq. (1) at every time step.

$$\Theta_{(y_{[t]})} = \sum_{t}^{T} \theta_{(y_t)} \tag{2}$$

The goal of the decoding algorithm (Eq. (3)) is to determine the word combination $\hat{y}$ that maximizes the score defined in Eq. (2). However, greedy search does not guarantee that selecting the highest probability word at each step generates the output sentence with the maximum score.

$$\hat{y} = \underset{y}{\operatorname{argmax}} \sum_{t=1}^{T} \Theta_{(y_{[t]})} \tag{3}$$

3.3.2. Beam search

Beam search improves the greedy search problem by generating the top-B predicted words at each time step.

Fig. 4 shows an example of a beam search with a beam size of 3. The algorithm selects the top three words with the highest probability in the first time step ($t = 1$). In the second time step, decoding is performed for each of the three selected words, and the algorithm selects the top three words among all three sets of probabilities.

In this example, the result from y_1^{beam2} was not selected, whereas the result from y_1^{beam1} and the other two words from the result of y_1^{beam3} were selected. The three words selected in the second step were input to the decoder in the third time step. This process was repeated until ⟨eos⟩ appears, and three final sentences are generated.

Beam search guarantees that the decoder generates better sentences than a greedy search. Beam search increases the accuracy scores using the beam size. However, the problem that generated sentences are too general or similar to each other still exists [14,36]. Sampling methods based on the probability distribution [30,33] or cumulative probability [34] have been proposed to solve the problem of beam search. Although sampling-based methods alleviate the problem of beam search to a certain extent, the diversity problem remains unsolved [6].

3.3.3. Diverse beam search

Diverse beam search (DBS) [14] uses an additional hyperparameter G (group size) in addition to the beam size. DBS divides the input data by the number of G and processes them with a beam search for each group. Thus, the previous beam search can be viewed as a diverse beam search with $G = 1$.

If G is equal to or greater than 2, a beam search of the beam with B' (Eq. (4)) is performed for each group.

$$B' = B \times \frac{1}{G} \tag{4}$$

Each group computes scores sequentially with respect to all previous groups using a similarity function, except the first group. The beam search computes the score for each candidate word using Eq. (5), which extends the decoding score from Eq. (1) by incorporating a similarity-based diversity term.

$$\theta_{\left(y^g_{\{b',t\}}\right)} = \sum_{t}^{T} \left\{ \theta_{\left(y^g_{\{b',t\}}\right)} + \sum_{i=1}^{g-1} \lambda \cdot Similarity\left(y^g_{\{b',t\}}, y^i_t\right) \right\}, \tag{5}$$

where b' denotes the beam size within each group, $y^g_{\{b',t\}}$ is the candidate word at time step t for group $g > 1$, drawn from the group-specific vocabulary V^g, and λ_g is a hyperparameter greater than zero that controls the strength of the diversity term.

For example, during the second time step of group 3′s beam search, candidate words are compared to the outputs of groups 1 and 2 at the same step using a similarity function. The resulting similarity scores are weighted by the diversity strength λ. These weighted scores are then added to the decoding score in Eq. (5).

The property shown in Eq. (5) is that the diversity with the previous group affects the prediction of the current output words in a weighted sum method. This property allows DBS to generate output sentences with high diversity compared to beam search.

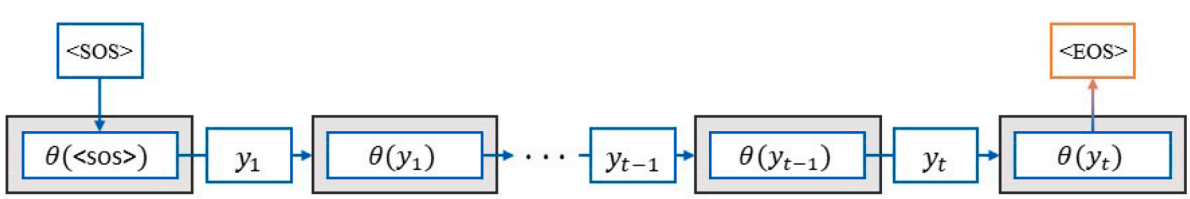


Fig. 3. Process of greedy search algorithm.

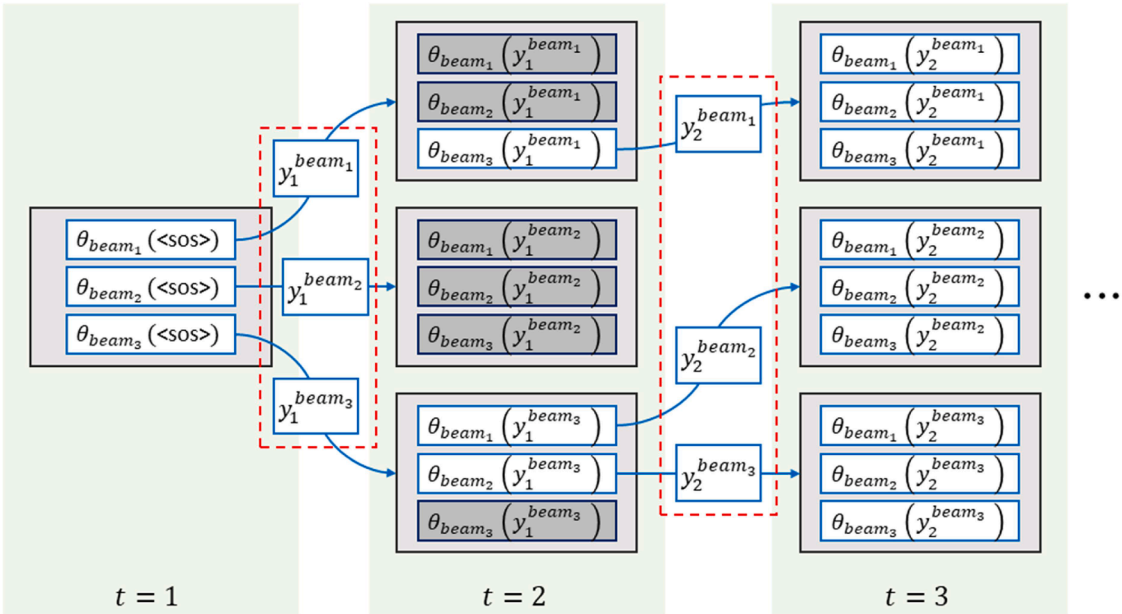


Fig. 4. Process of beam search algorithm ($B = 3$).

3.4. Problem description

In this section, we introduce the fundamental concepts of semantic diversity, meaningful variation, and lexical diversity. We then illustrate the limitations of existing decoding algorithms.

3.4.1. Basic concepts

Semantic diversity is variation in scene graph content across captions for the same image-i.e., differences in objects, attributes, or relations. We evaluate it with SPICE (scene graph level semantic content) and ALLSPICE, which combines SPICE with Self-CIDEr to balance semantic accuracy with within-image diversity (Section 5.1).

Meaningful variation is a stricter, pairwise interpretive criterion: two captions are meaningfully different only if at least one scene graph tuple (object, attribute, or relation) changes and the change is supported by the image (not a surface substitution such as synonyms or inflections, and not a major semantic error). We apply this criterion in qualitative analyses to count only content-changing differences.

Lexical diversity denotes surface-form variation and is used only as a comparison baseline; when we report a lexical baseline, we use Self-CIDEr.

Unless otherwise noted, "diversity" denotes semantic diversity. This distinction matters: DBS increases lexical diversity by penalizing Hamming overlap yet may not yield meaningful variation, whereas SDBS explicitly promotes semantic diversity via knowledge-graph similarity.

3.4.2. Limitations of existing approaches

Problem 1. Semantic diversity: DBS has limitations in terms of semantic diversity. One of the similarity functions used in DBS is the Hamming similarity (Eq. (6)).

$$HammingSimilarity(W_1,\ W_2) = \frac{1}{n}\sum_{i=1}^{N}\delta(W_1[i],\ W_2[i]), \tag{6}$$

where $W_1[i]$ and $W_2[i]$ denote the i-th characters of the words W_1 and W_2 respectively. The function $\delta(a,\ b)$ is an indicator function that returns 1 if a and b are equal, and 0 otherwise. N is the length of the longer word.

In Eq. (6), the numerator counts the number of matching characters in corresponding positions. Since this is a purely string-based method, DBS computes diverse similarity scores based on surface-level character matches without considering semantic meaning.

Problem 2. Computational efficiency: DBS measures the similarity of all words in the vocabulary sets. Then, DBS computes diversity scores using all similarities. However, words with low similarity scores can be judged as diverse words. The redundant and inefficient process must be improved, which is the diverse computation between words with low similarity.

Problem 3. Issues on hyperparameters: Adopting the beam size and group size in DBS exponentially increases the computational cost. Moreover, DBS has a functional dependency between beam size and group size, and assigning the same value to beam size and group size is recommended.

4. Method

In this section, we propose semantic diverse beam search (SDBS), a decoding algorithm that preserves semantic diversity by augmentation issues described in Section 3.4.

4.1. Semantic diversity score

In contrast to DBS, which relies on string-based similarity functions, our method computes semantic similarity using WuPalmer (WUP) similarity to measure the semantic similarity between words [37,38]. WUP Similarity utilizes the open graph WordNet [39,40], a lexical database of English words with semantic relations.

Eq. (7) describes WUP similarity. *Depth(x)* denotes the shortest path between the root and the word *x*. *LCS(a, b)* denotes the lowest common ancestor of *a* and *b*. For example, in Fig. 1(b), *LCS("COFFEE", "ESPRESSO")* returns *"Beverage"*. *LCS(a, b)* always returns a non-zero value, which ensures that the WUP similarity score is always between 0 and 1 [41,42].

$$Similarity(W_1,\ W_2) = \frac{2\ \times Depth(LCS(W_1,\ W_2))}{Depth(W_1) + Depth(W_2)} \tag{7}$$

In our decoding algorithm, semantic diversity scores are computed using Eq. (8). The diversity score is the sum of similarity between candidate words of current group g and the predicted words from previous groups ranging from group 1 to group $g-1$. The variable t denotes

the current time step, and b' denotes the beam size within each group. During the summation, each similarity value is scaled by the diversity strength λ. The final diversity score is then returned as a negative sum, which implies that the diversity score imposes a penalty on semantically similar words.

$$DiverseScore(y) = -\sum_{i=1}^{g-1} \lambda \cdot Similarity\left(y^{'g}_{\{b',t\}}, Y^{i}_{t}\right) \tag{8}$$

4.2. Similarity threshold

We define a similarity threshold to manage the issue of computational efficiency discussed in Section 3.4. If the semantic similarities between words are lower than the similarity threshold, our method does not proceed with further diversity computations for the words. Because the words with low semantic similarities are already diverse, our method excludes the words from calculating diversity scores.

We set the threshold to 0.8 to determine the degree of similarity. Using the similarity threshold, we reduced the computational cost and confirmed through experiments that adopting the threshold preserves accuracy and diversity.

4.3. Stratified top-k sampling over time steps

As an additional component, our method incorporates top-k sampling to filter out low probability words prior to semantic similarity evaluation. The top-k sampling enables low-cost decoding and ensures reasonable accuracy and diversity.

Assume two groups, g_1 and g_2, where decoding algorithms are applied to g_2. Diversity scores are computed between g_2 and g_1. The vocabulary sizes used are $|V_{g2}|$=100 and $|V_{g1}|$=400. Top-k sampling is set to $k = 10$. We conservatively assume that an 0.8 similarity threshold reduces the dataset size by half. The total computational cost is

$$Cost = \left|V_{g_2}\right| \times N_{similarity\ computation} \times N_{diverse\ score\ computation}, \tag{9}$$

where N denotes the number of similarity computations performed between a single word in V_{g2} and all words in V_{g1}.

In the previous approach, the total computational cost is $100 \times 400 \times 400 = 1.6 \times 10^7$, as each of the 100 words in V_{g2} is compared with all 400 words in V_{g1}, across all 400 entries in g_2. In contrast, our approach reduces the cost to $10 \times 400 \times 200 = 8 \times 10^5$, by applying top-k sampling that reduces similarity computations from 100 to 10, and a similarity threshold that reduces diversity score computations from 400 to 200. Top-k sampling enables our algorithm to compute diversity scores only for words with high probabilistic significance.

In addition to the top-k sampling, we stratified the top-k sampling over time steps. The time step is the spatial information of words in a sentence in NLP processing and is implicitly related to the part-of-speech of words. A sentence has various part-of-speech distributions over the positions of words (Fig. 5). The horizontal axis represents time steps. The vertical axis represents the average part-of-speech distributions of sentences of COCO dataset [43]. Therefore, we used stratified top-k sampling over time steps to adopt the distributional features of the text data.

4.4. Beam size normalization

We normalized the diversity score computation by dividing the score by the beam size. Normalization lowers the interaction between the beam size and accuracy. Normalization also lowers the functional dependency between the beam size and group size mentioned in Section 3.4. In our semantic diverse beam search algorithm, the beam size and group size values are assigned separately as independent hyperparameters.

4.5. Algorithm

Algorithm 1 is a pseudocode of the provided semantic diverse beam search algorithm (SDBS). The number of time steps is denoted as T. The number of groups is denoted as G; if the input vocabulary size is 100 and G is 5, then each of the 20 data points is assigned to the five groups.

Line 3 – 4. Beam search: The first group does not have other processed groups for comparison. The first group is computed by a normal beam search.

The hyperparameter τ is used as an early-stopping concept in all groups. If we practically determine a specific time step that converges to a certain accuracy, then we set τ as the time step to terminate SDBS and process beam search after τ time steps for a better computation cost.

Line 5 – 15. Semantic diverse beam search (SDBS): After the first

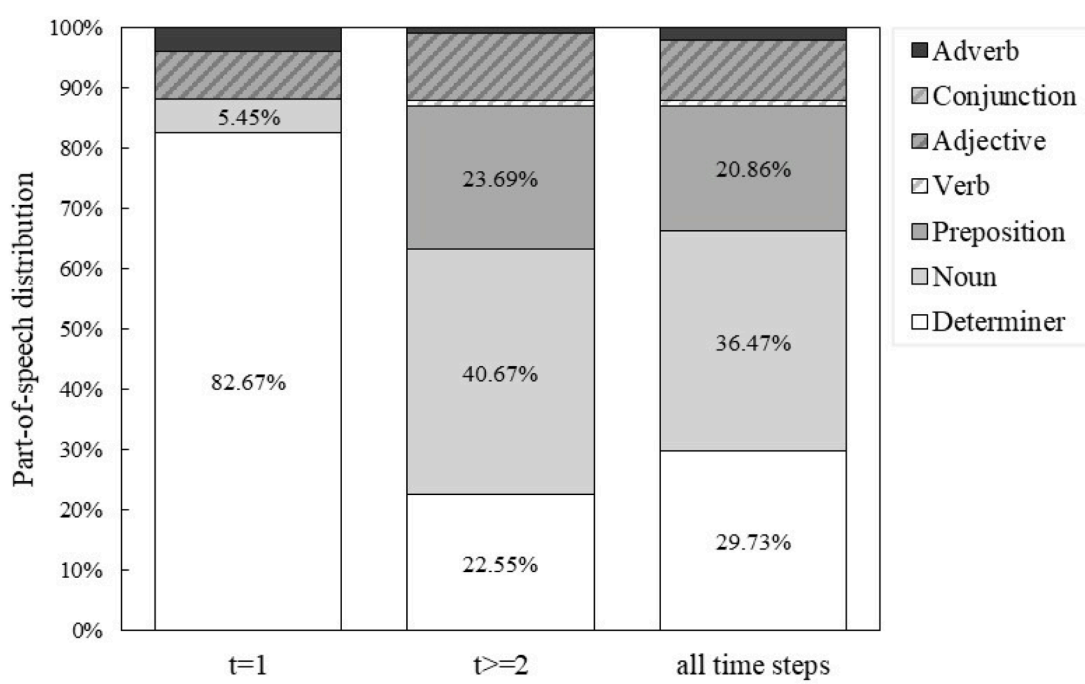


Fig. 5. Part-of-speech distributions over time steps.

Algorithm 1
Semantic diverse beam search pseudocode.

1 **for** t = 1, …, T **do**
2 **for** g = 1, …, G **do**
3 **if** g = 1 **or** t > τ **then**
4 $Y_t^g \leftarrow \text{BeamSearch}\left(y^{g}_{b',t}\right)$
5 **else**
6 $Y_t' \leftarrow \text{TopKSampling}(y_t, t)$
7 $\Phi_{Y_t'} \leftarrow \text{SemanticSimilarity}\left(y'^{g}_{b',t'},\ Y^{i}_{t}\right)$
8 **if** $\Phi_{Y_t'} \geq 0.8$ **then**
9 $diverse_score \leftarrow -\sum_{i=1}^{g-1} \lambda \cdot \Phi\left(y'^{g}_{b',t'}, Y_t^i\right)$
10 **else**
11 $diverse_score \leftarrow 0$
12 **end if**
13 $\Theta'\left(y'^{g}_{b',t}\right) \leftarrow \sum_t^T \{\theta(y'_{b',t}) + \text{Norm}(diverse_score)\}$
14 $Y_t^g \leftarrow \underset{y',b'}{\text{argmax}}\, \Theta'(y^{g}_{b',t})$
15 **end if**
16 **end**
17 **end**
18 **return best set** $\{Y_t^g\}$

group, the other groups are computed using SDBS.

Line 6. Stratified top-k sampling: Candidate words are pruned using top-k sampling with stratification on time steps.

Line 7. Semantic similarity: The pruned data of the current group are compared to the previously processed groups using a semantic similarity function; for example, group 4 is compared to groups 1, 2, and 3.

Line 8–12. Diversity score with similarity threshold: If the semantic similarity of the current word pair is equal to or greater than 0.8, a diversity score is computed with diversity strength λ. Otherwise, a diversity score is assigned zero, representing no penalty between the diverse words.

Line 13: Word probability adjustment: The diversity score is normalized by the beam size and added to word probabilities to assign penalties between similar word pairs.

Line 14: Sentence prediction: SDBS generates best sentences that contain the highest adjusted probabilities for each group.

Line 18. Output sentences: The algorithm returns all the predicted sentences. The number of sentences is B' $\times$ G (beam size in group $\times$ group size).

5. Experiments

Our experiment used the Common Objects in Context 2014 (COCO 2014) [43,44] image captioning dataset. The dataset consists of 164,100 images containing five caption text labels per image. COCO 2014 is a comprehensive dataset that contains diverse everyday scenes with objects in their natural contexts, including 80 object categories such as people, animals, vehicles, and household items. The dataset provides rich annotations for both object detection and image captioning tasks, making it a standard benchmark for evaluating image captioning models.

For our experiments, we utilized pre-trained Transformer and A2I2 (Attention to Attention with Image) [45] models that were trained on the COCO training dataset. These models serve as our base image captioning architectures, allowing us to evaluate the effectiveness of our SDBS decoding algorithm independently of the underlying model architecture.

We used a single machine with an Intel Xeon Gold 6134 3.2 GHz CPU, NVIDIA Quadro GP100, and 32GB RAM. The machine ran Ubuntu 20.04 LTS, Python 3.8.13, Pytorch-Cuda 11.6, cuDNN 8.4.0, and NLTK 3.7.

The experiment was conducted separately on each method implemented in our semantic diverse beam search algorithm (SDBS). The methods were evaluated based on diversity and accuracy. Finally, we compared the SDBS algorithm with the previous approach.

5.1. Evaluation metrics

To comprehensively evaluate the performance of image captioning models, this study adopts both traditional quantitative metrics and modern neural-based evaluation approaches, considering both accuracy and diversity aspects.

For traditional evaluation, we employ BLEU-4 [46] to evaluate direct word-level matching by measuring precision up to 4-grams, ROUGE-L [47] to measure structural similarity based on longest common subsequence, METEOR [48] to evaluate lexical and semantic consistency using stemming and synonym mapping, and CIDEr [49] to quantify consensus between generated and reference captions based on TF-IDF weighted n-gram similarity.

Semantic-based evaluation includes SPICE [50], which measures semantic consistency at the object, attribute, and relationship level by comparing captions at the scene graph level, and ALLSPICE [51], a composite metric that integrates SPICE-based semantic consistency with Self-CIDEr-based caption diversity to evaluate the balance between semantic accuracy and diversity.

For diversity assessment, we utilize Self-CIDEr [12], which quantifies caption generation diversity by averaging CIDEr similarity between multiple captions generated for the same image.

Additionally, we employ modern neural-based metrics including BARTScore++ [52], which distinguishes between major and minor errors using the MQM framework, and EAPrompt, a technique combining Chain-of-Thought reasoning with error analysis for human-like evaluation [53].

All evaluations were conducted using publicly available official scripts, with each decoding algorithm generating 5 captions per image. Detailed experimental methods for modern neural-based metrics are provided in Appendix A.

5.2. Semantic diversity score

5.2.1. Semantic similarity

Fig. 6 presents a comprehensive comparative evaluation of accuracy and diversity performance across varying similarity functions used in diversity score computations (beam size: 5, group size: 5). The horizontal axis represents diversity strength values (λ), while the vertical axis shows ALLSPICE scores that measure the balance between semantic accuracy and caption diversity.

The results reveal a fundamental difference in how string-based and semantic-based similarity methods respond to diversity constraints. SDBS with threshold consistently achieves superior performance, while SDBS without threshold exhibits severe performance degradation at higher λ values, demonstrating the critical importance of selective similarity computation in knowledge graph-based approaches.

5.2.2. Similarity threshold

Fig. 6 illustrates the fundamentally different computational requirements between string-based and semantic-based similarity approaches, revealing why indiscriminate semantic similarity computation can be counterproductive.

String-based DBS shows minimal response to threshold application due to the inherently noisy nature of Hamming similarity measurements. Since both similar and dissimilar words receive relatively arbitrary similarity scores, threshold filtering provides little meaningful benefit for lexical-based approaches.

In contrast, SDBS without threshold exhibits a critical flaw: while high-probability candidate words appropriately receive diversity penalties due to their semantic similarity to previous context, low-probability irrelevant words avoid penalties entirely due to their low semantic similarity scores. As diversity strength (λ) increases, these inappropriate words paradoxically gain higher selection probabilities, leading to semantically incoherent outputs and performance degradation.

The 0.8 similarity threshold elegantly resolves this issue by restricting diversity penalty computation to semantically related word pairs only. This ensures that diversity constraints operate within the space of reasonable candidates, preventing artificial promotion of irrelevant words while maintaining meaningful diversification among appropriate alternatives.

5.3. Stratified top-k sampling

Fig. 7 shows whether stratified top-k sampling is effective (beam size: 5, group size: 5, SDBS threshold: 0.8). The horizontal axis represents diversity strength values. The vertical axis represents the ALLSPICE scores. SDBS generates more accurate and diverse outputs than DBS. Moreover, SDBS with the top-k sampling method performs better. Fig. 7 shows representative stratified top-k sampling ratios, where the front-loaded ratio of 10:3 significantly outperforms balanced (3:3) and back-loaded (3:10) approaches. Based on these insights, we further tested additional front-loaded ratios and found that 9:2 achieved the best overall performance, which we adopted as our default configuration. The detailed analysis of parameter selection factors for stratified top-k sampling ratios and their impact on performance across different time steps is provided in Appendix B.

5.4. Beam size normalization

Fig. 8 illustrates the effectiveness of diversity score normalization (beam size:5, group size:5, SDBS threshold:0.8). The horizontal axis represents the beam size. The vertical axis represents the ALLSPICE scores. The results show that normalized SDBS generates more accurate and diverse output sentences.

An additional effect of normalization is the smoothing beam search curse problem [54]. The beam search curse problem is that a large beam size adversely affects the decoding performance. In Fig. 8, we observe that the normalized SDBS decreases the trade-off between decoding performance and beam size.

The previous approach recommended setting the beam size and group size to be equal. In contrast, SDBS with normalization demonstrates the functional independence between the beam size and group size. Table 2 shows the results obtained by varying the beam size and group size. For example, the bold score of 0.278 in the first line is the best score for a beam size of 5 with group 6. The best scores do not exist on the diagonal line, where the beam sizes and group sizes are equal. The

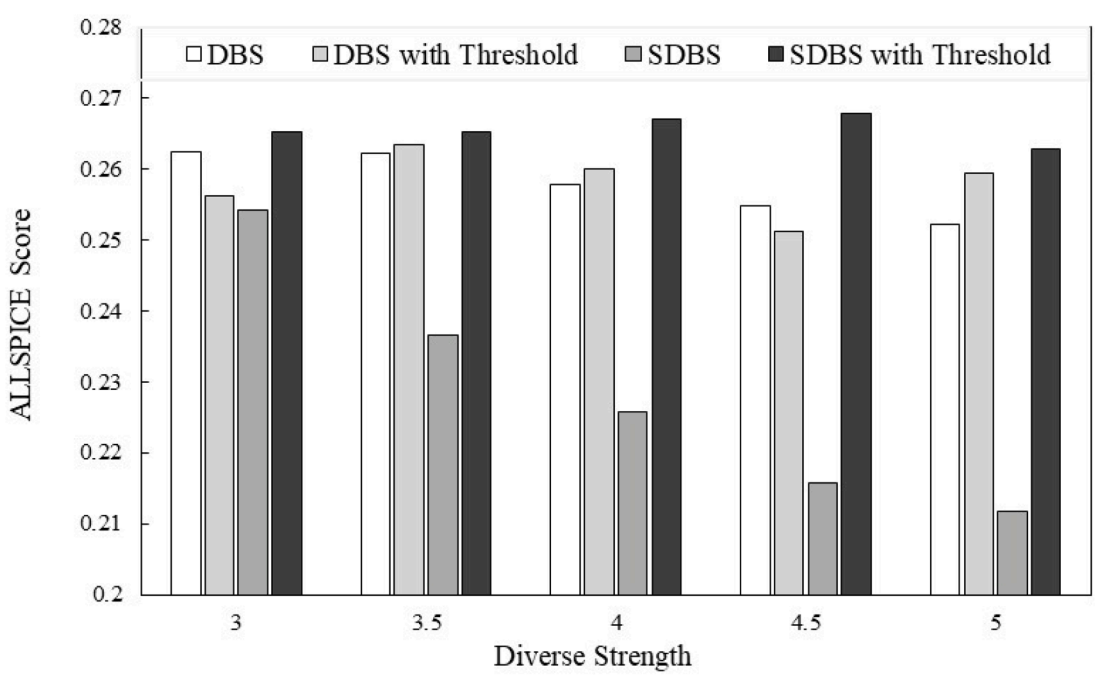


Fig. 6. Impact of similarity threshold on semantic diversity across diversity strengths (λ). Applying a 0.8 threshold enables SDBS to maintain stable ALLSPICE scores while preventing the performance degradation observed in without threshold implementations at high λ values.

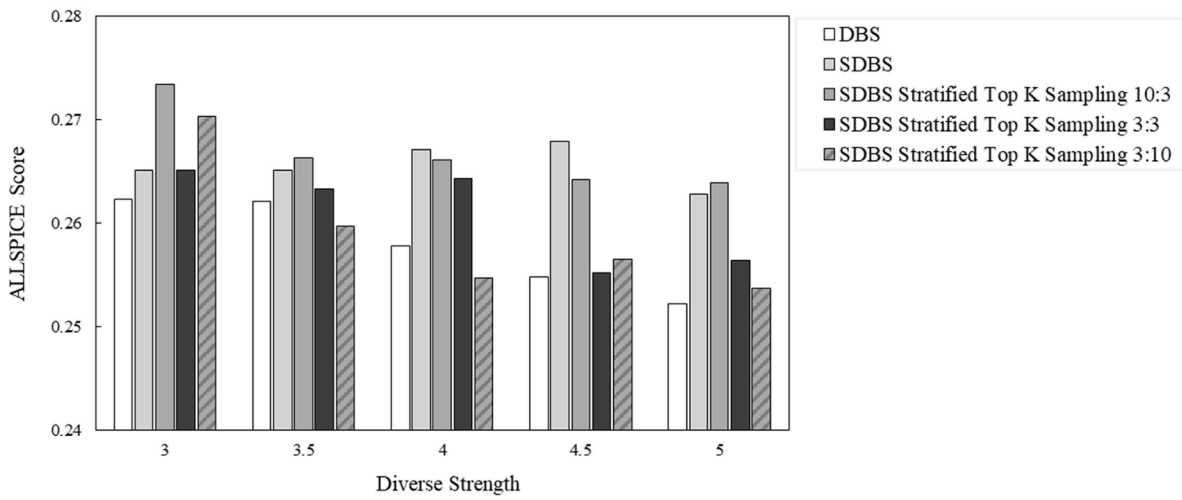


Fig. 7. Effectiveness of stratified top-k sampling with different ratios. A 10:3 ratio assigns more of the top-k pool to the initial decoding steps, leading to higher ALLSPICE scores by combining early-step accuracy with later-step diversity.

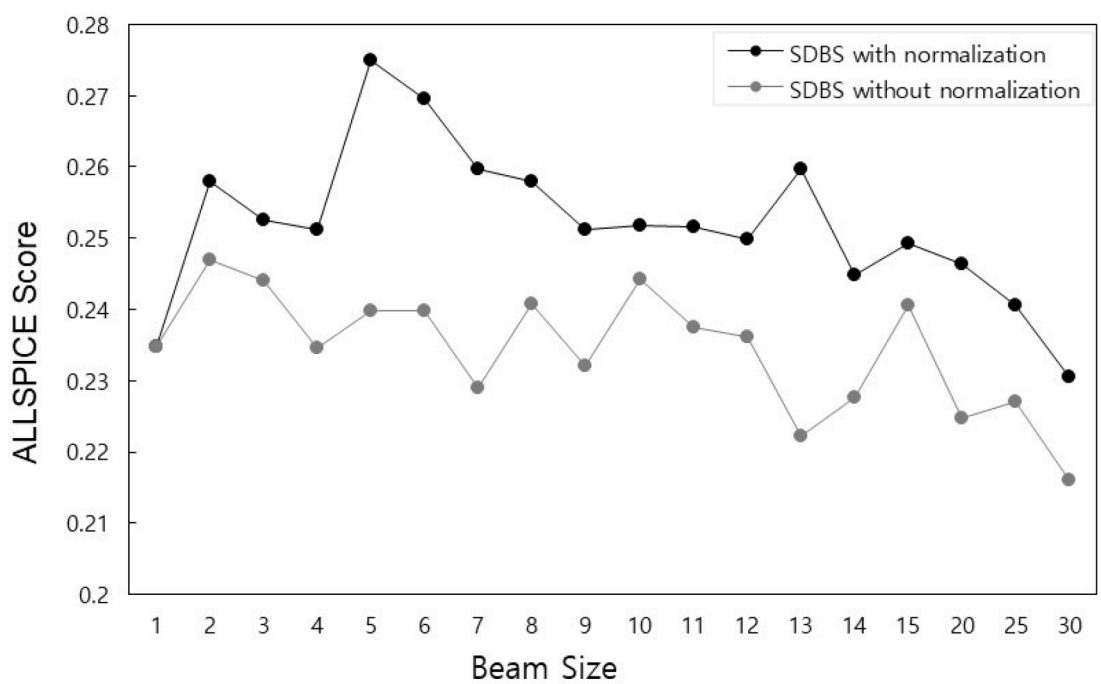


Fig. 8. Comparison of SDBS with and without beam size normalization. Normalization mitigates the beam search curse effect, maintaining consistent performance across beam sizes and enabling independent tuning of beam and group size parameters.

results show that our approach can be used with various beam size and group size combinations.

5.5. Early-Stop strategy in SDBS over time steps

To examine whether applying SDBS to all time steps is always efficient, we experimented with an early-stop strategy (SDBS τ@N) that applies SDBS only to the first N decoding steps before switching to standard beam search. As shown in Fig. 9, $\tau = 13$ consistently matched the performance of full SDBS while reducing computational cost, which is why we adopted this setting. In contrast, smaller or larger τ values sometimes improved performance but also led to drops depending on the diversity strength λ, so $\tau = 13$ offered the best trade-off without any performance degradation.

5.6. Ablation study

To quantitatively analyze the contribution of each component of our proposed SDBS (Semantic Diverse Beam Search) decoding algorithm to image captioning performance, we conducted an ablation study. The experiments were performed with a fixed Transformer-based image captioning model, where major components of SDBS were individually removed or modified to observe changes in performance metrics.

According to the results shown in Table 3, removing the Semantic Similarity Threshold resulted in a decrease of 0.029 in ALLSPICE score and 0.026 in SPICE score, confirming that overall accuracy and semantic diversity both deteriorated. This demonstrates the effectiveness of the threshold technique in eliminating unnecessary computations for word pairs with low similarity and inducing semantics-based diversification.

Furthermore, when Diversity Strength (λ) was removed (fixed to $\lambda =$

Table 2
Evaluation of SDBS with varying beam size and group size.

Evaluation metric		group 5	group 6	group 7	group 8	group 9	group 10
ALLSPICE	beam 5	0.275	**0.278**	0.275	0.268	0.263	0.263
	beam 6	**0.270**	0.267	0.270	0.268	0.261	0.259
	beam 7	0.260	0.256	**0.260**	0.255	0.250	0.247
	beam 8	**0.258**	0.256	0.253	0.247	0.249	0.248
	beam 9	0.251	0.255	0.254	0.255	0.256	**0.261**
	beam 10	0.252	0.258	**0.262**	0.260	0.255	0.252
SPICE	beam 5	0.255	0.261	0.263	0.265	0.266	**0.266**
	beam 6	0.240	0.245	0.253	0.257	0.260	**0.260**
	beam 7	0.238	0.240	0.244	0.247	0.252	**0.255**
	beam 8	0.236	0.237	0.241	0.243	0.244	**0.244**
	beam 9	0.234	0.240	0.242	0.243	0.244	**0.252**
	beam 10	0.238	0.243	0.248	0.249	0.254	**0.256**
Self-CIDEr	beam 5	**0.832**	0.825	0.817	0.818	0.813	0.811
	beam 6	0.813	**0.813**	0.810	0.808	0.807	0.802
	beam 7	**0.825**	0.824	0.818	0.819	0.814	0.811
	beam 8	**0.815**	0.813	0.810	0.809	0.807	0.802
	beam 9	**0.804**	0.802	0.801	0.798	0.792	0.795
	beam 10	0.797	**0.797**	0.795	0.794	0.793	0.789

Table 3
Ablation study of SDBS components using Transformer-based image captioning model.

Configuration	Evaluation metric						
	ALLSPICE	SPICE	CIDEr	Self-CIDEr	ROUGE-L	BLEU-4	METEOR
Full (SDBS)	**0.275**	**0.255**	**1.429**	**0.832**	**0.623**	**0.367**	**0.337**
w/o Semantic Threshold	0.246	0.229	1.308	0.818	0.596	0.339	0.317
w/o Diversity Strength (1)	0.245	0.229	1.354	0.704	0.616	0.373	0.328
w/o Top-k sampling	0.265	0.244	1.357	0.814	0.610	0.338	0.332
$\tau = 0$ (no early stop)	**0.275**	**0.255**	**1.429**	**0.832**	**0.623**	**0.367**	**0.337**

1), the Self-CIDEr score decreased by 0.128, suggesting that λ is a key factor in distinguishing semantic differences between candidate sentences. In other words, SDBS maintains informational richness by adjusting penalties for semantically similar words through λ.

When stratified top-k sampling was removed, both ALLSPICE and CIDEr scores declined (by -0.010 and -0.072 respectively), demonstrating that the candidate word sampling technique reflecting part-of-speech distributions across time steps positively affects both diversity and sentence quality.

Meanwhile, the experiment extending SDBS application to all decoding steps ($\tau = 0$) showed nearly identical results to the original τ=13 setting. This suggests that the influence of diversity constraints may be limited in the latter part of sentence generation, supporting that the τ=13 setting is appropriate considering both computational efficiency and performance.

Table 4 presents the computational efficiency analysis of each SDBS component. The full SDBS configuration achieves the fastest execution time (5.203 s), demonstrating that our design choices optimize both quality and efficiency. Removing the semantic threshold increases execution time by 16 % (6.031 s) due to unnecessary diversity computations for semantically distant word pairs. Similarly, removing top-k sampling results in a 19 % increase (6.211 s) as the algorithm must process larger candidate sets. The absence of early stopping ($\tau = 0$) causes the most significant overhead with a 24 % increase (6.462 s), confirming that applying SDBS to all decoding steps provides minimal benefit while substantially increasing computational cost.

Table 4
Computational efficiency of SDBS components through ablation analysis.

Configuration	Time (second)
Full (SDBS)	**5.203**
w/o Semantic Threshold	6.031
w/o Diversity Strength (1)	**5.203**
w/o Top-k sampling	6.211
$\tau = 0$ (no early stop)	6.462

Interestingly, removing diversity strength (fixing $\lambda = 1$) maintains the same execution time as full SDBS (5.203 s), as this modification only affects the weighting calculation rather than the computational complexity. These results validate that SDBS's efficiency gains primarily stem from intelligent candidate filtering (threshold and top-k sampling) and strategic early termination, rather than from simplifying the diversity scoring mechanism itself.

Comprehensively, the ablation studies in Tables 3 and 4 reveal that each component of the proposed SDBS works synergistically to achieve three objectives simultaneously: semantic diversity, accuracy, and computational efficiency. The performance metrics (Table 3) show consistent degradation when components are removed, while the timing analysis (Table 4) paradoxically shows that the full SDBS runs faster than most ablated versions. This counterintuitive result, where adding components improves speed demonstrates that SDBS achieves high performance through intelligent design that eliminates redundant computations rather than through computational brute force. The organic integration among components validates our design principle of balancing generation quality with practical deployment efficiency.

5.6.1. *Component analysis through concrete examples*

To illustrate how each SDBS component prevents system breakdown, we present concrete examples demonstrating failure modes when key techniques are removed:

Without Semantic Threshold: When generating captions for an image containing a dog, SDBS without threshold incorrectly promotes semantically unrelated words like "building" or "sky" simply because they have low semantic similarity to previous outputs. The 0.8 threshold ensures diversity constraints operate only within semantically meaningful candidate spaces, preventing artificial promotion of irrelevant terms.

Without Diversity Strength (λ) Control: Fixed λ=1 fails to distinguish between critical semantic differences and minor variations. For example, when describing a "running dog," the system cannot differentiate between semantically similar alternatives like "sprinting canine" (meaningful diversity) and "racing puppy" (surface variation). The adaptive λ=3.0 setting enables appropriate penalty weighting, ensuring meaningful semantic distinctions are preserved while preventing trivial lexical variations.

Without Stratified Top-k Sampling: Standard top-k sampling ignores positional linguistic patterns, leading to inappropriate word selection across sentence positions. For instance, without stratified sampling, the system might generate "beautiful sleeps peacefully" (adjective as subject), whereas the 9:2 stratified ratio respects part-of-speech distributions, maintaining both diversity and linguistic naturalness.

These examples demonstrate that SDBS components work synergistically to prevent specific failure modes rather than simply improving numerical scores, validating our design principles through concrete linguistic evidence.

5.7. Practical implementation guidelines

For practitioners implementing SDBS in real-world applications, we provide empirically-validated hyperparameter selection guidelines based on our comprehensive experiments:

Similarity Threshold: Start with 0.8 as the default threshold across different domains and datasets. This value effectively balances computational efficiency with semantic precision. Values below 0.7 introduce excessive noise from semantically unrelated words, while values above 0.9 may be too restrictive for capturing meaningful diversity.

Diversity Strength (λ): For most applications, λ=3.0 provides optimal balance between semantic diversity and accuracy. Increase λ to 4.0–5.0 when prioritizing diversity over accuracy (e.g., creative applications), or decrease to 1.0–2.0 when strict factual accuracy is paramount (e.g., medical or scientific domains). Our experiments show that λ values outside the 1.0–5.0 range either provide insufficient diversity control or introduce semantic instability.

Stratified Top-k Sampling: Use a front-loaded split by default (9:2), which in our comprehensive testing outperformed all other ratios including the 10:3 shown in Fig. 7. While Fig. 7 demonstrates the superiority of front-loaded approaches, the optimal 9:2 ratio provides the best balance between early-step precision and computational efficiency. It consistently outperforms balanced/reversed splits by prioritizing semantically heavy early tokens. For short captions ($\leq$ 10 tokens) consider 7:3; for longer ($\geq$ 15) keep 9:2 (or 10:3 if validated on your data). Bayesian optimization sometimes prefers 4:3, showing that moderate candidate pools can be optimal; treat it as a dataset-specific alternative rather than the default.

Early Stopping (τ): Define τ as the number of initial decoding steps where SDBS is applied. Thus, $\tau = T$ is full SDBS and $\tau = 0$ is beam-only. A practical default is $\tau \approx 0.8$ T (e.g., $\tau = 13$ in our COCO setup), which preserves ALLSPICE while cutting semantic-similarity computations substantially.

5.8. Overall comparison

This section presents a comprehensive performance comparison between the proposed SDBS and existing decoding algorithms. The comparative evaluation employed empirically validated parameter settings (beam size: 5, group size: 5, SDBS threshold: 0.8, Diversity strength λ=3.0, SDBS stratified top-k sampling ratio: 9:2, SDBS normalization applied, τ=13) derived from our systematic parameter analysis shown in Fig. 6-9, evaluating performance from traditional metrics to modern neural-based evaluation approaches. Additionally, to validate our manual hyperparameter selection approach, we conducted Bayesian optimization experiments detailed in Appendix D. The automated optimization results (λ=2.882, threshold=0.892, sampling ratio=4:3) closely matched our manual settings, confirming the near-optimality of our empirically-derived configurations.

5.8.1. Performance and efficiency analysis

Table 5 demonstrates that SDBS achieved consistent superiority across both models in traditional evaluation metrics. For the Transformer-based model, SDBS recorded the highest performance across all existing methods with SPICE (0.255), CIDEr (1.429), BLEU-4 (0.367), METEOR (0.337), and ALLSPICE (0.275). The A2I2 model also achieved the best results in semantic-based metrics including ALLSPICE (0.259), SPICE (0.236), and METEOR (0.322). ALLSPICE improved by 38 % from 0.199 to 0.275 in the Transformer compared to standard Beam Search, demonstrating an effective balance between accuracy and diversity.

Table 6 further validates SDBS excellence in modern automatic

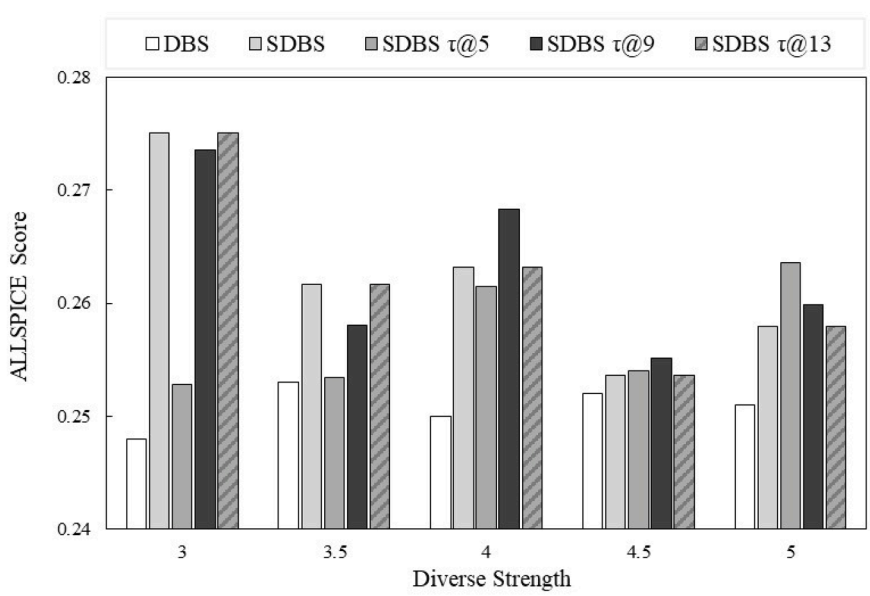


Fig. 9. Performance comparison of SDBS under different early-stop strategies. Setting $\tau = 13$ (SDBS for steps 1–13, then beam search) yields the same ALLSPICE scores as full SDBS across the tested diversity strengths, whereas other τ values show minor variations.

Table 5
Comparative evaluations of accuracy and diversity.

Model - Dataset	Evaluation metric								
- Decoding algorithm	ALLSPICE	SPICE	CIDEr	Self-CIDEr	ROUGE-L	BLEU-4	METEOR	BARTScore++	LLM-Eval
Transformer – COCO									
- BS	0.199	0.191	1.124	0.768	0.540	0.244	0.291	−1.266	−12.58
- Top-k	0.259	0.247	1.312	0.825	0.594	0.290	0.314	−1.277	−12
- Nucleus	0.239	0.237	1.328	**0.861**	0.595	0.310	0.314	−1.323	−15.34
- DBS	0.270	0.246	1.353	0.823	0.612	0.343	0.322	−1.167	−9.66
- SDBS	**0.275**	**0.255**	**1.429**	0.832	**0.623**	**0.367**	**0.337**	**−1.159**	**−9.04**
A2I2 - COCO									
- BS	0.170	0.170	1.061	0.600	0.504	0.195	0.260	−1.292	−16.92
- Top-k	0.238	0.228	1.345	**0.808**	0.568	0.285	0.296	−1.303	−16.28
- Nucleus	0.227	0.225	1.367	0.631	0.598	0.335	0.309	−1.496	−20.24
- DBS	0.255	0.233	**1.407**	0.798	0.594	**0.345**	0.316	−1.201	−11.34
- SDBS	**0.259**	**0.236**	1.369	0.778	**0.598**	0.339	**0.322**	**−1.186**	**−10.44**

Table 6
Comparative evaluations of decoding algorithm execution time.

Model/Decoding algorithm	Time (second)
Transformer/BS	1.422
Transformer/DBS	5.486
Transformer/SDBS	**5.203**
A2I2/BS	0.687
A2I2/DBS	3.916
A2I2/SDBS	**3.135**

evaluation metrics. SDBS outperformed all existing methods in BARTScore++ with Transformer (−1.159) and A2I2 (−1.186), and consistently achieved the best results in LLM-based evaluation with Transformer (−9.04) and A2I2 (−10.44). This suggests that SDBS demonstrates robust performance not only in traditional evaluation metrics but also in modern evaluation approaches requiring deep semantic understanding.

Table 6 computational efficiency analysis shows that SDBS reduced execution time compared to existing DBS despite performing complex semantic similarity-based computations (Transformer: 5.486→5.203 s, A2I2: 3.916→3.135 s). This indicates that the similarity threshold and top-k sampling techniques effectively eliminated unnecessary computations.

To further validate the generalizability of SDBS across different dataset characteristics and domains, we conducted additional experiments on Flickr30K [55,56] and preliminary experiments on GoodNews [57], a news domain dataset. As detailed in Appendix C, SDBS consistently outperformed baseline methods across both datasets, achieving superior ALLSPICE scores and other semantic metrics despite their distinct characteristics including Flickr30K's narrative style and GoodNews requirement for contextual knowledge. This comprehensive validation spanning general and specialized domains strengthens the evidence that SDBS represents a robust solution for semantic diversity enhancement.

5.8.2. Model-Specific analysis and comprehensive assessment

The high-expressivity Transformer decoder demonstrated consistent superiority across all metrics, confirming that semantic similarity-based control strategies operate precisely within rich representation spaces. The structurally simpler A2I2 showed slightly lower results in some n-gram based metrics (CIDEr, BLEU-4), which can be interpreted as a phenomenon where penalty application removes correct expressions due to narrow probability distribution characteristics, while still maintaining superiority in semantic-based metrics. This phenomenon reflects the fundamental difference between semantic accuracy (captured by SPICE) and surface-level lexical matching (measured by n-gram-based metrics), with detailed analysis provided in Appendix E.

The high scores achieved by Nucleus sampling in Self-CIDEr result from stochastic randomness maximizing superficial expression dispersion, demonstrating that SDBS's conservative diversity control strategy aligns with the design intention of prioritizing information density and sentence quality.

Comprehensively, SDBS achieved consistent performance improvements across both structurally different models and demonstrated superiority across diverse evaluation paradigms from traditional metrics to modern neural-based evaluations. Particularly, we confirmed that the effects of semantic similarity-based control strategies are maximized as model expressivity increases.

5.8.3. Qualitative analysis

Fig. 10 shows output sentences using decoding algorithms. We evaluated human preference for the output sentences. The beam search (BS) algorithm generates redundant and too general sentences with low diversity scores. Although DBS generates diverse sentences compared to BS, DBS generates some semantically wrong diverse sentences (highlighted in red in Fig. 10). SDBS generates more semantically diverse sentences than other algorithms. Our analysis reveals that SDBS effectively leverages inter-object relationships for sentences to diversely describe images (highlighted in blue in Fig. 10).

6. Conclusion

This work introduced Semantic Diverse Beam Search (SDBS), which fundamentally reframes the diversity challenge in neural text generation by operating in semantic rather than surface lexical space. Our findings demonstrate that incorporating knowledge graph-based semantic understanding during decoding produces outputs that are both more meaningfully diverse and higher quality than existing approaches, challenging the conventional assumption that diversity and accuracy are inherently at odds.

The comprehensive evaluation validated SDBS's effectiveness across multiple dimensions. On COCO-2014, SDBS achieved superior performance across traditional metrics (SPICE: 0.255, CIDEr: 1.429, ALLSPICE: 0.275) while maintaining computational efficiency through similarity thresholding and stratified sampling techniques. The consistent improvements observed across both Transformer and A2I2 architectures demonstrate that semantic diversity control can enhance various neural generation frameworks without architectural modifications. Cross-dataset validation on Flickr30K (Appendix C) further confirmed the generalizability of our approach across diverse dataset characteristics.

Crucially, SDBS operates as a training-free, inference-only method that can be immediately applied to any pre-trained autoregressive captioning model. As established by prior research [14,33,35–37], decoding algorithms demonstrate consistent effectiveness across diverse autoregressive frameworks due to their shared dependence on probability distributions rather than internal model architectures. This universality makes SDBS broadly applicable without requiring retraining or

Fig. 10. Output sentences using decoding algorithms.

model-specific adaptations, addressing practical deployment concerns in real-world applications.

Our experimental methodology underwent rigorous validation through multiple approaches. Comprehensive ablation studies confirmed the necessity of each SDBS component (Table 3), while systematic parameter analysis across different configurations established optimal hyperparameter settings (Fig. 6-9). Additional validation through Bayesian optimization (Appendix D) confirmed the near-optimality of our empirically-derived configurations, and cross-dataset experiments (Appendix C) demonstrated consistent performance improvements across different dataset characteristics. The systematic evaluation across traditional and modern neural-based metrics established SDBS as a reliable solution for semantic diversity enhancement.

Despite these achievements, several limitations present opportunities for future advancement. The current reliance on WordNet for semantic similarity may limit performance in domain-specific or multilingual contexts where specialized knowledge graphs would be more appropriate. Additionally, while our evaluation demonstrates

effectiveness across COCO-2014, Flickr30K, and preliminary GoodNews experiments, comprehensive validation across additional benchmark datasets (Visual Genome, nocaps) with full-scale domain-specific evaluations would further establish broader applicability. Future work could also explore the theoretical relationship between semantic and surface diversity to yield principled frameworks for understanding when and why semantic approaches outperform lexical methods.

As neural text generation systems increasingly shape human-computer interaction, methods like SDBS that balance quality with meaningful diversity will be essential for creating AI systems capable of engaging with the full richness of human language. The ability to produce semantically diverse yet factually accurate outputs represents a fundamental step toward more controllable and interpretable AI systems, particularly crucial for high-stakes applications requiring both creativity and reliability. This capability advances the broader goal of developing trustworthy artificial intelligence systems that can meaningfully contribute to human expression and understanding.

Funding

The present Research has been conducted by the Research Grant of Kwangwoon University in 2022.

CRediT authorship contribution statement

HyungSun Na: Writing – review & editing, Writing – original draft, Supervision, Methodology, Conceptualization. **Hee-Gook Jun:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Formal analysis, Conceptualization. **Jinhyun Ahn:** Writing – review & editing, Methodology, Conceptualization. **Dong-Hyuk Im:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

We adapted the Error Analysis Prompting (EAPrompt) technique, originally developed for machine translation evaluation, to assess image captioning quality. The original EAPrompt framework was designed to evaluate translation quality using Chain-of-Thought reasoning combined with error analysis following the MQM (Multidimensional Quality Metrics) framework. The original template for machine translation evaluation is illustrated in Fig. 11. We modified this approach to evaluate the semantic accuracy and fluency of generated image captions.

We employed GPT-4.1-mini as the evaluation model for all EAPrompt-based assessments. The model was configured to operate in JSON mode to ensure structured output parsing and eliminate the need for regular expression-based error extraction, thereby improving reliability and consistency of the evaluation process. An example of the adapted prompt text and the expected JSON response schema is provided in Fig. 12.

The original EAPrompt prompts were specifically designed for machine translation evaluation, focusing on source-target language pair assessment. We systematically adapted these prompts for image captioning by:

1. **Context Modification**: Replacing source-target language comparison with image-caption correspondence evaluation
2. **Error Category Adaptation**: Modifying translation-specific error types to caption-relevant categories (semantic accuracy, object detection, spatial relationships, fluency)
3. **Evaluation Framework**: Maintaining the MQM-inspired error severity classification while adapting criteria to visual-textual alignment

Following the MQM framework adaptation, we implemented a two-tier error severity system:

- **Major Errors (−5 points)**: Semantic distortions, incorrect object identification, misrepresented spatial relationships
- **Minor Errors (−1 point)**: Grammatical issues, stylistic inconsistencies, minor fluency problems

The final score for each caption is calculated as the average of individual error-adjusted scores across all evaluation criteria.
The adapted prompt template for image captioning evaluation follows this structure:

```
Source: <SRC>
Reference: <REF>
Translation: <TGT>
Based on the given source and reference, identify the major and minor errors in this translation. Note that Major errors refer
to actual translation or grammatical errors, and Minor errors refer to smaller imperfections, and purely subjective opinions
about the translation.

<RESPONSE1>

Count the number of major and minor errors identified in your last response and compute the final score for this translation.
Deduct 5 points for each major error. Deduct 1 point for each minor error. If the translation has no errors, its score will be 0.

<RESPONSE2>
```

Fig. 11. Original EAPrompt Template for Machine Translation Evaluation.

```
Source: <IMG>
Reference: <REF>
Caption: <GEN>

Based on the given image (as the source) and the reference caption, identify the major and minor errors in the generated
caption.

Note that:
- Major errors refer to factual mistakes, hallucinated objects, missing essential elements, or grammatical issues that distort
the meaning.
- Minor errors refer to smaller imperfections such as slight omissions, minor rewordings, or purely subjective opinions that do
not significantly affect the understanding.

Please return your response in the following JSON format:
{
 "major_errors": [
   {
    "type": "<error_type>",
    "description": "<brief description>"
   }
 ],
 "minor_errors": [
   {
    "type": "<error_type>",
    "description": "<brief description>"
   }
 ]
}
```

Fig. 12. Adapted EAPrompt Template for Image Captioning Evaluation.

Appendix B

To systematically investigate the optimal stratified top-k sampling parameters, we initially tested three representative sampling ratios (10:3, 3:3, 3:10) shown in Fig. 13-15, representing fundamentally different strategies: front loading (10:3) prioritizes high probability candidates in early time steps while maintaining minimal diversity in later steps, balanced sampling (3:3) maintains equal computational allocation across time steps as a neutral baseline, and back loading (3:10) emphasizes diversity in later time steps while using fewer candidates initially. These experiments confirmed the superiority of front loaded approaches. Subsequently, we conducted detailed testing of various front loaded ratios and found that 9:2 consistently achieved the highest performance across all metrics, leading to its selection as the default configuration for our main experiments. Fig. 13 shows the SPICE score distribution across First Top-K and Remainder Top-K parameter combinations, where the 10:3 region consistently achieves higher semantic accuracy. Fig. 14 presents Self-CIDEr analysis, demonstrating that front-heavy sampling maintains adequate diversity while preserving quality. Fig. 15 displays ALLSPICE results, confirming that the 10:3 ratio optimally balances accuracy and diversity. The systematic analysis across these three metrics reveals that front-heavy sampling (10:3) outperforms both balanced (3:3) and back-heavy (3:10) approaches, validating our hypothesis that prioritizing high-quality candidates in early decoding steps while maintaining controlled diversity in later steps yields superior overall performance, thereby supporting the effectiveness of the 10:3 ratio demonstrated in Fig. 7.

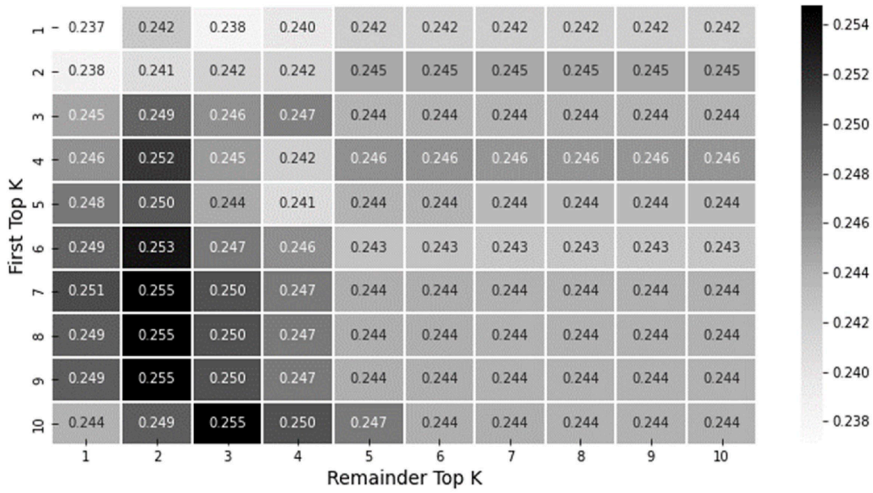


Fig. 13. SPICE Score Analysis for Stratified Top-k Sampling Parameter Optimization.

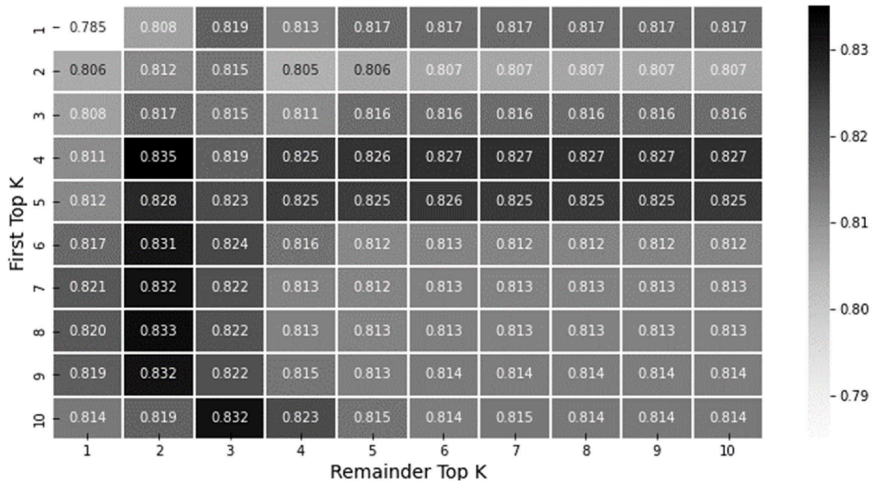


Fig. 14. Self-CIDEr Score Analysis for Stratified Top-k Sampling Parameter Optimization.

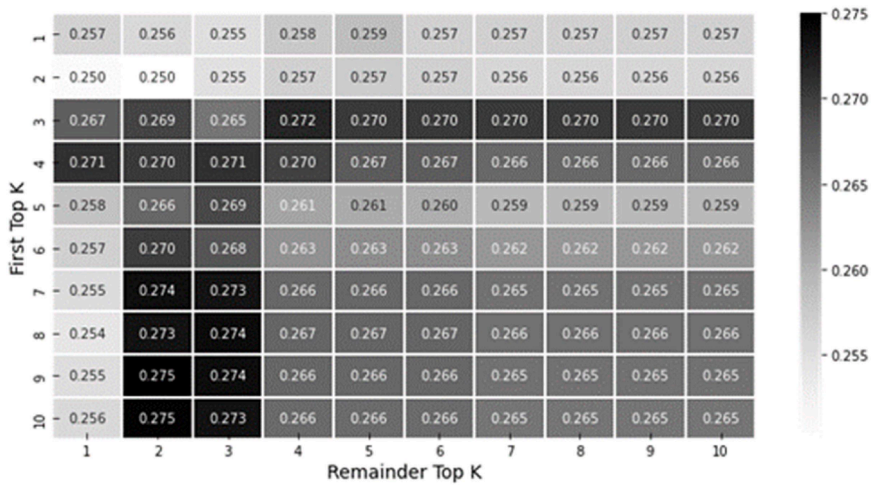


Fig. 15. ALLSPICE Score Analysis for Stratified Top-k Sampling Parameter Optimization.

Appendix C

To validate SDBS generalizability across different dataset characteristics and domains, we conducted additional experiments on Flickr30K [55,56] and GoodNews [57] datasets.

Flickr30K consists of 31,000 images with five human-written captions each, presenting simpler scenes and more narrative caption styles compared to COCO-2014. Using ablation-guided parameter optimization, we achieved optimal configurations of (beam size: 5, group size: 5, threshold: 0.9, λ=3.5, sampling ratio: 9:3, τ=13) for Transformer and (threshold: 0.8, λ=4.5, sampling ratio: 7:2) for A2I2. As shown in Table 7, SDBS consistently outperformed baseline methods, with Transformer achieving ALLSPICE (0.173), SPICE (0.170), and CIDEr (0.633).

Table 7
Comparative evaluations on Flickr30K dataset.

Model - Dataset	Evaluation metric								
- Decoding algorithm	ALLSPICE	SPICE	CIDEr	Self-CIDEr	ROUGE-L	BLEU-4	METEOR	BARTScore++	LLM-Eval
Transformer - Flickr30K									
- BS	0.122	0.108	0.478	0.621	0.430	0.113	0.178	−1.720	−12.20
- Top-k	0.144	0.150	0.596	0.869	0.478	0.110	0.229	−1.669	−11.42
- Nucleus	0.129	0.144	0.541	**0.925**	0.475	0.125	0.225	−1.865	−13.28
- DBS	0.155	0.158	0.599	0.844	0.485	0.122	0.227	−1.709	−11.00
- SDBS	**0.173**	**0.170**	**0.633**	0.816	**0.494**	**0.158**	**0.236**	**−1.663**	**−10.16**
A2I2 - Flickr30K									
- BS	0.125	0.094	0.379	0.564	0.431	0.101	0.160	−1.730	−12.60
- Top-k	0.139	0.144	0.543	0.893	0.484	0.107	0.217	−1.747	−12.88
- Nucleus	0.113	0.127	0.432	**0.951**	0.442	0.068	0.215	−2.058	−15.72
- DBS	0.155	0.147	0.614	0.766	0.504	0.156	0.229	−1.694	−11.10
- SDBS	**0.158**	**0.149**	**0.623**	0.815	**0.509**	**0.164**	**0.231**	**−1.666**	**−10.34**

GoodNews represents a specialized domain challenge as a news image captioning dataset requiring contextual knowledge and named entity understanding. To evaluate SDBS effectiveness in specialized domains, we conducted a preliminary study on 100 sampled images augmented with human annotated references to enable diversity metric computation. We performed Bayesian optimization following the procedure described in Appendix D, yielding parameters adapted for this domain (λ=4.94, threshold=0.85, sampling ratio=14:6), with the notably higher sampling ratio compared to COCO's 9:2 suggesting that news content benefits from larger candidate pools to accommodate diverse named entities, temporal references, and contextual terms. While absolute scores are substantially lower than COCO/Flickr30K (ALLSPICE 0.085 vs. 0.275/0.173) due to the dataset's complexity, which requires references to specific events, people, and information not visible in images, SDBS achieved the best balance of accuracy and diversity across all metrics (Table 8), demonstrating robustness even in this challenging specialized domain where both factual precision and semantic diversity are crucial.

Table 8
Comparative evaluations on GoodNews dataset.

Model - Dataset	Evaluation metric						
- Decoding algorithm	ALLSPICE	SPICE	CIDEr	Self-CIDEr	ROUGE-L	BLEU-4	METEOR
Transformer- GoodNews							
- BS	0.071	0.079	0.136	0.577	0.240	0.021	0.088
- Top-k	0.083	0.081	0.130	0.903	0.290	0.023	0.122
- Nucleus	0.073	0.076	0.120	**0.949**	0.288	0.023	0.114
- DBS	0.075	0.082	0.141	0.842	0.301	0.034	0.131
- SDBS	**0.085**	**0.085**	**0.155**	0.860	**0.308**	**0.039**	**0.134**

These validations across datasets demonstrate that SDBS maintains effectiveness across varying scales, domains (from general to news domains), and evaluation settings (from full datasets to limited samples). The successful application of both ablation based optimization (Flickr30K) and Bayesian optimization (GoodNews) confirms the robustness of our hyperparameter tuning approaches across different domains. Notably, SDBS achieved consistent improvements even without specialized knowledge graphs for specific domains, providing evidence for broad applicability. While comprehensive evaluations on larger specialized samples with multiple references annotated by humans remain future work, these preliminary results on 100 samples demonstrate SDBS's potential for semantic diversity enhancement even in challenging specialized domains.

Appendix D

We applied Gaussian Process–based Bayesian optimization to automatically tune SDBS's key hyperparameters (diversity strength λ, similarity threshold, and stratified top-k sampling ratio). Optimization was initialized with 12 random samples and proceeded for 50 iterations, totaling 62 model evaluations, and all runs converged within these 50 iterations. The comparison of manual settings and Bayesian optimization results is summarized in Table 9.

Table 9
Comparison of Manual Settings and Bayesian optimization Results.

Method	diversity strength(λ)	threshold	stratified top-k sampling ratio	ALLSPICE
Manual (Greedy)	3.0	0.8	9:2	0.275
Bayesian optimization	2.882066	0.891964	4:3	0.275133

Key observations:

1. **Reproduction of expert settings**: The optimized values ($\lambda = 2.882$ vs. 3.00; sim = 0.892 vs. 0.800) closely mirror manual tuning.
2. **Optimal candidate pool size**: A moderate top-k configuration (4 : 3) outperformed larger pools, demonstrating that limiting candidate diversity can enhance both efficiency and ALLSPICE score.

The validation of manual settings showed that the optimized values for diversity strength (2.882 vs. 3.0) and threshold (0.892 vs. 0.800) were broadly consistent with the manually tuned parameters. While the automated process favored a slightly higher similarity threshold, the proximity of the results validates that the initial manual configuration was located within a near-optimal range. Interestingly, Bayesian optimization converged to 4:3, which differs from our manually-tuned optimal of 9:2. This suggests that when all parameters are jointly optimized (λ=2.882, threshold=0.892), a more moderate distribution becomes effective, though our extensive ablation studies confirmed 9:2 as the best choice when using λ=3.0 and threshold=0.8.

Appendix E

We observed opposing results for the A2I2 model with the SDBS method, which achieved a higher SPICE score but lower scores on the n-gram-based metrics, CIDEr and BLEU-4. These opposing results are attributed to the metrics' distinct evaluation criteria. The SPICE score improved because the SDBS method generated captions with higher descriptive specificity. Conversely, the CIDEr and BLEU-4 scores decreased because their evaluation is based on literal n-gram overlap, a method that penalizes the use of semantically valid but more specific terms (e.g., "silver cat", "tarmac" vs "runway") when they do not appear in the more general reference captions. Representative examples of this effect are shown in Table 10 and Fig. 16.

Table 10
Representative examples comparing DBS and SDBS outputs for the A2I2 model.

Example	Method	Generated Caption	BLEU-4	CIDEr	SPICE
1	A2I2-DBS	there is a cat that is sitting on a chair	4.347e-6	1.936	0.273
	A2I2-SDBS	the silver cat is sitting on the chair	1.096e-12	1.418	0.308
2	A2I2-DBS	a group of planes parked on a runway	5.721e-5	0.914	0.261
	A2I2-SDBS	an airport tarmac with a lot of planes	8.034e-9	0.849	0.400

(a) Example 1 (b) Example 2

Fig. 16. Example image demonstrating A2I2 model caption generation comparison.

Reference Captions Example 1:

- A gray tiger cat sitting at a wooden table on a chair.
- A grey cat sitting in chair next to a table.
- A cat sitting in a chair pulled up to a table.
- A cat sitting in a chair at a table with a book on it.
- A cat sitting in a chair by a table.

Reference Captions Example 2:

- An airport filled with planes sitting on tarmacs.
- The view of runway from behind the windows of airport.
- a truck driving towards some planes parked on the runway.
- Planes on a wet tarmac unloading at arrival gates.
- Window view from the inside of airplanes, baggage carrier and tarmac.

These phenomena are illustrated by comparing SDBS captions with the provided reference captions. In Example 1, the SDBS caption ("the silver cat is sitting on the chair") prefers "silver" over "gray/grey," increasing attribute specificity. In Example 2, SDBS selects "tarmac," a precise term that appears less frequently across the references compared with "runway," which reduces n-gram overlap and thus lowers CIDEr/BLEU-4 despite being semantically accurate. This aligns with the design goal of prioritizing semantic fidelity (SPICE) over lexical conformity.

We also observed opposing results on the Flickr30K dataset with the Transformer model, where SDBS yields higher SPICE but lower CIDEr and BLEU-4 than DBS. This is again due to the metrics' different criteria: SDBS tends to choose more specific or relation-explicit words (e.g., lake, splashing, rugby), which improves scene-graph semantics (SPICE) while reducing literal n-gram overlap with narrative references. Representative cases are in Table 11 and Fig. 17.

Table 11
Representative examples comparing DBS and SDBS outputs for Flickr30K (Transformer).

Example	Method	Generated Caption	BLEU-4	CIDEr	SPICE
1	Trans-DBS	a boy playing in the water	0.160	1.058	0.206
	Trans-SDBS	the little boy splashing in a lake	0.141	1.016	0.267
2	Trans-DBS	three men tackle a man in red	0.218	1.214	0.171
	Trans-SDBS	three rugby players bring down a red player	0.173	1.026	0.188

(a) Example 1 (b) Example 2

Fig. 17. Example image demonstrating Transformer model caption generation comparison.

Reference Captions Example 1:

- a small child in water with a splash encircling him while the white clouds float over the mountains
- a little boy is playing in the water with a beautiful sunset and mountains in the background
- a young boy plays in the water with the mountains in the background
- a child is splashing in the water
- a boy playing in a lake

Reference Captions Example 2:

- three men wearing the same black and yellow uniform are blocking another man in a red uniform
- three rugby players in black strips are pulling the player in red to the ground
- three players bring a team opponent to the ground
- three men tackle someone from the opposite team
- a group of young men in sports uniforms from two opposing teams are in vigorous play on a grassy field

In Flickr30K Example 1, SDBS generates "the little boy splashing in a lake" which adds both action verb "splashing" and specific location "lake", improving semantic completeness (SPICE: 0.267 vs 0.206) while reducing n-gram overlap with references that use different phrasings. Example 2 shows SDBS selecting "rugby players" over generic "men", providing sport-specific context that enhances semantic accuracy (SPICE: 0.188 vs 0.171) but diverges from reference vocabulary. The Flickr30K dataset's narrative style amplifies this semantic-lexical divide, as its longer descriptive captions offer more opportunities for meaningful variation that SDBS exploits through semantic similarity control.

We vary the early-stop parameter $\tau \in \{\text{FULL}, 13, 9, 5\}$ on three images to test whether semantic decisions consolidate early. Straightforward scenes (Examples 1–2) converge by $\tau \approx 13$-Self-CIDEr is unchanged for FULL/$\tau = 13$ and deviates only when τ is set very small, whereas the ambiguous case (Example 3) shows a modest but consistent spread. Hence we adopt $\tau = 13$ as our default, which matches full SDBS while reducing semantic similarity computations, consistent with Section 5.5 (Fig. 9). See Table 12 and Fig. 18 for details.

Table 12
Early-stop analysis (SDBS τ@N): applying SDBS for the first N steps and then switching to beam search.

Example	τ	Generated Caption	Self-CIDEr	CIDEr	SPICE
1	FULL	a couple of ducks swimming on top of a lake	0.769	0.834	0.263
	13	a couple of ducks swimming on top of a lake	0.769	0.834	0.263
	9	there are two ducks that are swimming in the water	0.651	1.529	0.263
	5	two ducks floating on top of a body of water	0.772	1.596	0.060
2	FULL	the young woman is posing for a picture	0.915	0.470	0.294
	13	the young woman is posing for a picture	0.915	0.470	0.294
	9	the young woman is posing for a picture	0.915	0.470	0.294
	5	the lady is posing with her hair in her hair	0.838	0.049	0.200
3	FULL	an image of an old photo of a painting	0.950	0.737	0.231
	13	an image of an old photo of a painting	0.950	0.737	0.231
	9	an image of an old photo of a painting	0.950	0.737	0.231
	5	this picture shows an image of a pitcher and a tea kettle	0.902	0.167	0.160

(a) Example 1 (b) Example 2 (c) Example 3

Fig. 18. Example images demonstrating early stopping impact on SDBS caption generation.

Reference Captions Example 1:

- Two ducks floating together on a body of water.
- Two ducks are swimming in the green colored pond.
- Two Canadian geese swim in a green pond.
- Two ducks swim in a pond with green water.
- Two swam swimming next to each other on a lake.

Reference Captions Example 2:

- An image of a very cute girl with face piercings.
- Painting of girl that has multiple piercings on face
- A young brunette woman with multiple face piercings.
- A young woman with green eyes and piercings all over her face.
- A drawing of a young woman with many facial piercings.

Reference Captions Example 3:

- A painting of a table with fruit on top of it.
- Painting of oranges, a bowl, candle, and a pitcher
- a painting of fruit and a candle with a vase
- A painting of a candlestick holder with a candle, several pieces of fruit and a vase, with a gold frame around the painting.
- A painting that has a gold frame on it.

For semantically straightforward images (Examples 1–2), τ variations yield identical or highly similar captions with stable Self-CIDEr, indicating that semantic decisions converge early in decoding. In Example 1, Self-CIDEr is identical for FULL and $\tau = 13$ (0.769), dips at $\tau = 9$ (0.651), and returns near baseline at $\tau = 5$ (0.772). Example 2 remains flat through FULL/$\tau = 13$/$\tau = 9$ (0.915) with a drop only at $\tau = 5$ (0.838). In contrast, the ambiguous image (Example 3) shows a narrower but consistent spread (Self-CIDEr 0.902–0.950), confirming that extended SDBS influences semantics mainly when scenes are challenging. Together with Section 5.5 (Fig. 9), these results support $\tau = 13$ as the default: it captures essential semantic diversity in early steps while avoiding unnecessary computation once captions have largely converged.

Comprehensively, these analyses confirm that SDBS functions as designed: prioritizing semantic diversity over superficial lexical conformity. The observed "performance degradation" in n gram metrics actually represents successful semantic diversification, generating more informative and specific descriptions that may not match reference vocabulary but accurately capture visual content. This approach centered on semantics aligns with the goal of producing genuinely diverse yet accurate captions rather than superficial lexical variations.

Data availability

the MS COCO images dataset is licensed under a Creative Commons Attribution 4.0

References

[1] H. Sharma, M. Agrahari, S.K. Singh, M. Firoj, R.K. Mishra, Image captioning: a comprehensive survey, in: Proc. Int. Conf. Power Electron. IoT Appl. Renew. Energy Control (PARC), IEEE, 2020, pp. 325–328, https://doi.org/10.1109/PARC49193.2020.236619.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (2015) 211–252, https://doi.org/10.1007/s11263-015-0816-y.

[3] S. Kim, T.H. Jeon, I. Rhiu, J. Ahn, D.-H. Im, Semantic scene graph generation using RDF model and deep learning, Appl. Sci. 11 (2021) 826, https://doi.org/10.3390/app11020826.

[4] D.H. Im, G.D. Park, Linked tag: image annotation using semantic relationships between image tags, Multimed. Tools Appl. 74 (2015) 2273–2287, https://doi.org/10.1007/s11042-014-1855-z.

[5] Y. Ming, N. Hu, C. Fan, F. Feng, J. Zhou, H. Yu, Visuals to text: a comprehensive review on automatic image captioning, IEEE/CAA J. Autom. Sinica 9 (2022) 1339–1365, https://doi.org/10.1109/JAS.2022.105734.

[6] H. Meng, X. He, Z. Chen, F. Zhou, IFDID: information Filter upon Diversity-Improved Decoding for Diversity-Faithfulness Tradeoff in NLG, arXiv preprint (2022), https://doi.org/10.48550/arXiv.2210.13829. arXiv:2210.13829.

[7] J. Shi, Y. Li, S. Wang, Partial off-policy learning: balance accuracy and diversity for human-oriented image captioning, in: Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2021, pp. 2187–2196, https://doi.org/10.1109/ICCV48922.2021.00219.

[8] Z. Sun, S. Huang, H. Wei, X. Dai, J. Chen, Generating diverse translation by manipulating multi-head attention, in: Proc. AAAI Conf. Artif. Intell. 34, 2020, pp. 8976–8983, https://doi.org/10.1609/aaai.v34i05.6429.

[9] X. Yue, X.F. Zhang, Z. Yao, S. Lin, H. Sun, CliniQG4QA: generating diverse questions for domain adaptation of clinical question answering, in: Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM), IEEE, 2021, pp. 580–587, https://doi.org/10.1109/BIBM52615.2021.9669300.

[10] Q. Chen, C. Deng, Q. Wu, Learning distinct and representative modes for image captioning, in: Adv. Neural Inf. Process. Syst. (NeurIPS) 35, 2022, pp. 9472–9485 [Online]Available, https://proceedings.neurips.cc/paper_files/paper/2022/hash/3d77c6dcc7f143aa2154e7f4d5e22d68-Abstract-Conference.html.

[11] Q. Wang, J. Wan, A.B. Chan, On diversity in image captioning: metrics and methods, IEEE Trans. Pattern Anal. Mach. Intell. 44 (2022) 1035–1049, https://doi.org/10.1109/TPAMI.2020.3013834.

[12] Q. Wang, A.B. Chan, Describing like humans: on diversity in image captioning, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2019, pp. 4195–4203, https://doi.org/10.1109/CVPR.2019.00432.

[13] S. Welleck, A. Bertsch, M. Finlayson, H. Schoelkopf, A. Xie, G. Neubig, I. Kulikov, Z. Harchaoui, From decoding to meta-generation: inference-time algorithms for large language models, Trans. Mach. Learn. Res. (TMLR) (2024) [Online]. Available, https://openreview.net/forum?id=eskQMcIbMS.

[14] A. Vijayakumar, M. Cogswell, R. Selvaraju, Q. Sun, S. Lee, D. Crandall, D. Batra, Diverse beam search for improved description of complex scenes, in: Proc. AAAI Conf. Artif. Intell. 32, 2018, pp. 7371–7379, https://doi.org/10.1609/aaai.v32i1.12340.

[15] M. Zhang, Z. Li, G. Fu, M. Zhang, Syntax-enhanced neural machine translation with syntax-aware word representations, in: Proc. Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. (NAACL-HLT), 2019, pp. 1151–1161, https://doi.org/10.18653/v1/N19-1118.

[16] K. Palasundram, N.M. Sharef, N.A. Nasharuddin, K.A. Kasmiran, A. Azman, Sequence to sequence model performance for education chatbot, Int. J. Emerg. Technol. Learn. 14 (2019) 56–68, https://doi.org/10.3991/ijet.v14i24.12187.

[17] K. Qian, Z. Yu, Domain adaptive dialog generation via meta learning, in: Proc. 57th Annu. Meeting Assoc. Comput. Linguist. (ACL), 2019, pp. 2639–2649, https://doi.org/10.18653/v1/P19-1253.

[18] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: Proc. Int. Conf. Learn. Represent. (ICLR), 2015, https://doi.org/10.48550/arXiv.1409.0473.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Adv. Neural Inf. Process. Syst. (NeurIPS) 30, 2017, pp. 5998–6008, https://doi.org/10.48550/arXiv.1706.03762.

[20] Q. You, H. Jin, Z. Wang, C. Fang, J. Luo, Image captioning with semantic attention, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 4651–4659, https://doi.org/10.1109/CVPR.2016.503.

[21] R. Biswas, M. Barz, D. Sonntag, Towards explanatory interactive image captioning using top-down and bottom-up features, beam search and re-ranking, Künstl. Intell. 34 (2020) 571–584, https://doi.org/10.1007/s13218-020-00679-2.

[22] M. Stefanini, M. Cornia, L. Baraldi, S. Cascianelli, G. Fiameni, R. Cucchiara, From Show to Tell: a Survey on Deep Learning-based Image Captioning, IEEE Trans. Pattern Anal. Mach. Intell. 45 (2023) 539–559, https://doi.org/10.1109/TPAMI.2022.3148210.

[23] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: a neural image caption generator, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 3156–3164, https://doi.org/10.1109/CVPR.2015.7298935.

[24] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, Y. Bengio, Show, attend and tell: neural image caption generation with visual attention, in: Proc. Int. Conf. Mach. Learn. (ICML), PMLR, 2015, pp. 2048–2057, https://doi.org/10.48550/arXiv.1502.03044.

[25] S.J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, V. Goel, Self-critical sequence training for image captioning, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 7008–7024, https://doi.org/10.1109/CVPR.2017.131.

[26] X. Zhu, L. Li, J. Liu, H. Peng, X. Niu, Captioning transformer with stacked attention modules, Appl. Sci. 8 (2018) 739, https://doi.org/10.3390/app8050739.

[27] L. Guo, J. Liu, X. Zhu, P. Yao, S. Lu, H. Lu, Normalized and geometry-aware self-attention network for image captioning, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 10327–10336, https://doi.org/10.1109/CVPR42600.2020.01034.

[28] Y. Luo, J. Ji, X. Sun, L. Cao, Y. Wu, F. Huang, C.W. Lin, R. Ji, Dual-level collaborative transformer for image captioning, in: Proc. AAAI Conf. Artif. Intell. 35, 2021, pp. 2286–2293, https://doi.org/10.1609/aaai.v35i3.16328.

[29] Z. Wang, J. Yu, A.W. Yu, Z. Dai, Y. Tsvetkov, Y. Cao, SimVLM: simple visual language model pretraining with weak supervision, in: Proc. Int. Conf. Learn. Represent. (ICLR), 2022, https://doi.org/10.48550/arXiv.2108.10904.

[30] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, OpenAI Technical Report, 2019 [Online]. Available, https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[31] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: Adv. Neural Inf. Process. Syst. (NeurIPS) 33, 2020, pp. 1877–1901, https://doi.org/10.48550/arXiv.2005.14165.

[32] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (2020) 1–67 [Online]. Available, https://jmlr.org/papers/v21/20-074.html.

[33] A. Fan, M. Lewis, Y. Dauphin, Hierarchical neural story generation, in: Proc. 56th Annu. Meeting Assoc. Comput. Linguist. (ACL), 2018, pp. 889–898, https://doi.org/10.18653/v1/P18-1082.

[34] A. Holtzman, J. Buys, L. Du, M. Forbes, Y. Choi, The curious case of neural text degeneration, in: Proc. Int. Conf. Learn. Represent. (ICLR), 2020, https://doi.org/10.48550/arXiv.1904.09751.

[35] Y. Su, N. Collier, Contrastive Search Is What You Need For Neural Text Generation, in: Trans. Mach. Learn. Res. (TMLR), 2023 [Online]. Available, https://openreview.net/forum?id=GbkWw3jwL9.

[36] R. Leblond, J.B. Alayrac, L. Sifre, M. Pislar, J.B. Lespiau, I. Antonoglou, K. Simonyan, O. Vinyals, Machine translation decoding beyond beam search, in: Proc. Conf. Empirical Methods Nat. Lang. Process. (EMNLP), 2021, pp. 8410–8434, https://doi.org/10.18653/v1/2021.emnlp-main.662.

[37] Z. Wu, M. Palmer, Verbs semantics and lexical selection, in: Proc. 32nd Annu. Meeting Assoc. Comput. Linguist. (ACL), Association for Computational Linguistics, 1994, pp. 133–138, https://doi.org/10.3115/981732.981751.

[38] P. Sharma, R. Tripathi, V.K. Singh, R.C. Tripathi, Automated patents search through semantic similarity, in: Proc. Int. Conf. Comput. Commun. Control (IC4), IEEE, 2015, pp. 1–5, https://doi.org/10.1109/IC4.2015.7375692.

[39] G.A. Miller, WordNet: a lexical database for English, Commun. ACM 38 (1995) 39–41, https://doi.org/10.1145/219717.219748.

[40] T. Pedersen, S. Patwardhan, J. Michelizzi, WordNet:similarity – measuring the relatedness of concepts, in: Proc. AAAI Conf. Artif. Intell., 2004, pp. 25–29, https://doi.org/10.3115/1614025.1614037.

[41] J. Matsuoka, Y. Lepage, Ambiguity spotting using WordNet semantic similarity in support to recommended practice for software requirements specifications, in: Proc. Int. Conf. Natural Lang. Process. Knowl. Eng. (NLPKE), 2011, pp. 479–484, https://doi.org/10.1109/NLPKE.2011.6138247.

[42] E.J. Sari, Y. Priyadi, R.R. Riskiana, Implementation of semantic textual similarity between requirement specification and use case description using WUP method (case study: sipjabs application), in: Proc. IEEE World AI IoT Congr. (AIIoT), 2022, pp. 681–687, https://doi.org/10.1109/AIIoT54504.2022.9817311.

[43] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, C.O.C.O. Microsoft, Common objects in context, in: Proc. Eur. Conf. Comput. Vis. (ECCV), Springer, 2014, pp. 740–755, https://doi.org/10.1007/978-3-319-10602-1_48.

[44] J. Kasai, K. Sakaguchi, L. Dunagan, J. Morrison, R.L. Bras, Y. Choi, N.A. Smith, Transparent human evaluation for image captioning, in: Proc. Conf. North Am. Chapter Assoc. Comput. Linguist. (NAACL), 2022, pp. 3464–3478, https://doi.org/10.18653/v1/2022.naacl-main.254.

[45] L. Huang, W. Wang, J. Chen, X.-Y. Wei, Attention on Attention for Image Captioning, in: Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2019, pp. 4634–4643, https://doi.org/10.1109/ICCV.2019.00473.

[46] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: a method for automatic evaluation of machine translation, in: Proc. 40th Annu. Meeting Assoc. Comput. Linguist. (ACL), 2002, pp. 311–318, https://doi.org/10.3115/1073083.1073135.

[47] C.Y. Lin, ROUGE: a package for automatic evaluation of summaries, in: Proc. ACL Workshop Text Summarization Branches Out, 2004, pp. 74–81.

[48] S. Banerjee, A. Lavie, METEOR: an automatic metric for MT evaluation with improved correlation with human judgments, in: Proc. ACL Workshop Intrinsic Extrinsic Eval. Measures Mach. Transl. Summarization, 2005, pp. 65–72.

[49] R. Vedantam, C.L. Zitnick, D. Parikh, CIDEr: consensus-based image description evaluation, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 4566–4575, https://doi.org/10.1109/CVPR.2015.7299087.

[50] P. Anderson, B. Fernando, M. Johnson, S. Gould, SPICE: semantic propositional image caption evaluation, in: Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 382–398, https://doi.org/10.1007/978-3-319-46454-1_24.

[51] R. Luo, G. Shakhnarovich, Analysis of diversity-accuracy tradeoff in image captioning, in: Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW), 2019, https://doi.org/10.48550/arXiv.2002.11848.

[52] Q. Lu, L. Ding, L. Xie, K. Zhang, D.F. Wong, D. Tao, Toward human-like evaluation for natural language generation with error analysis, in: Proc. 61st Annu. Meeting Assoc. Comput. Linguist. (ACL), 2023, pp. 5892–5907, https://doi.org/10.18653/v1/2023.acl-long.324.

[53] Q. Lu, B. Qiu, L. Ding, K. Zhang, T. Kocmi, D. Tao, Error analysis prompting enables human-like translation evaluation in large language models, in: Findings Assoc. Comput. Linguist. (ACL), 2024, pp. 8801–8816, https://doi.org/10.18653/v1/2024.findings-acl.520.

[54] C. Meister, T. Vieira, R. Cotterell, If beam search is the answer, what was the question?, in: Proc. Conf. Empirical Methods Nat. Lang. Process. (EMNLP), 2020, pp. 2173–2185, https://doi.org/10.18653/v1/2020.emnlp-main.170.

[55] P. Young, A. Lai, M. Hodosh, J. Hockenmaier, From image descriptions to visual denotations: new similarity metrics for semantic inference over event descriptions, Trans. Assoc. Comput. Linguist. 2 (2014) 67–78, https://doi.org/10.1162/tacl_a_00166.

[56] M. Hodosh, P. Young, J. Hockenmaier, Framing image description as a ranking task: data, models and evaluation metrics, J. Artif. Intell. Res. 47 (2013) 853–899, https://doi.org/10.1613/jair.3994.

[57] B. Biten, L. Gomez, M. Rusinol, D. Karatzas, Good news, everyone! Context driven entity-aware captioning for news images, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2019, pp. 12466–12475, https://doi.org/10.1109/CVPR.2019.01275.