# Thinking from abstract to embodiment: multi-leveled planning with LLMs aids in metaphor generation

Anonymous ACL submission

## Abstract

The emergence of Large Language Models (LLMs) has expanded the possibilities of zero-shot metaphor generation, but it often lacks interpretability and control, which are essential for meeting user needs. To tackle this problem, we draw inspiration from Conceptual Metaphor Theory by breaking the generation process into multiple intermediate steps. This approach views the creation of metaphors as a planning problem, emphasizing a progression from abstract concepts to embodied experiences. To address the challenge of navigating the extensive space of thoughts involved in metaphor generation, we incorporates feedback to refine the exploration strategy. Our proposed method can adjust the exploration in both depth and breadth directions to align with user needs, which are expressed as human criteria and integrated into the thread of thoughts. Evaluations of our model on an open test set demonstrate that it outperforms two types of baseline models for metaphor generation. Examples of metaphor generation with explanations show that our heuristic planning enables the discovery of more unique and effective intermediate thoughts.

## 1 Introduction

Metaphor generation plays a crucial role in enhancing the quality of language generation and fostering creative writing. Existing methods, such as those utilizing BART(Lewis et al., 2020) and T5(Raffel et al., 2020), typically rely on training data as these models are trained for supervised learning. However, these methods struggle when faced with a scarcity of parallel corpus.

Recently, Large Language Models (LLMs) have shown promising results in metaphor generation by capturing the richness and subtlety of human metaphorical expression, revealing new connections between the source and target domains (Prystawski et al., 2023). Despite this progress, LLMs tend to have an unpredictable tendency in candidate selection, which makes it difficult to control the generation process to align with user needs, posing challenges in practical applications.

Since metaphor generation is fundamentally a cognitive problem, we can draw upon human cognition to learn how to generate metaphors that can be controlled during the process. The cognitive view of metaphor has long been advocated by Black (Black, 1962) and Ortony (Ortony, 1980). Cognitive theorists support the idea of metaphors as structured thought process (Kövecses, 2005). Conceptual Metaphor Theory (Lakoff and Johnson, 2008) emphasizes the cognitive re-conceptualization of one idea or conceptual domain in terms of another. It views metaphors as structured thought mechanisms that enhance our understanding of abstract concepts. The theory asserts that our comprehension of abstract notions arises from metaphorical mappings rooted in tangible experiences and physical interactions with our environment. Therefore, we break the generation process into multi-leveled intermediate steps, starting from brainstorming abstract concepts and progressing towards exploring more concrete vehicles, ultimately leading to tangible experiences and bodily interactions with the world.

We view metaphor generation as a planning problem and introduce a multi-leveled metaphor tree-of-thoughts (MMToT) approach. To facilitate the multi-leveled thinking scaffold, we utilize the Tree of Thoughts (ToT) (Yao et al., 2023) in LLM planning. To address the challenge of navigating the vast thought space involved in metaphor generation, we propose a heuristic method for MMToT (HMMToT) that adjusts the exploration strategy based on various evaluation feedback, as illustrated in Figure 1.

Experimental results demonstrate that HMMToT achieves the highest average scores on most criteria in evaluations. Furthermore, HMMToT can

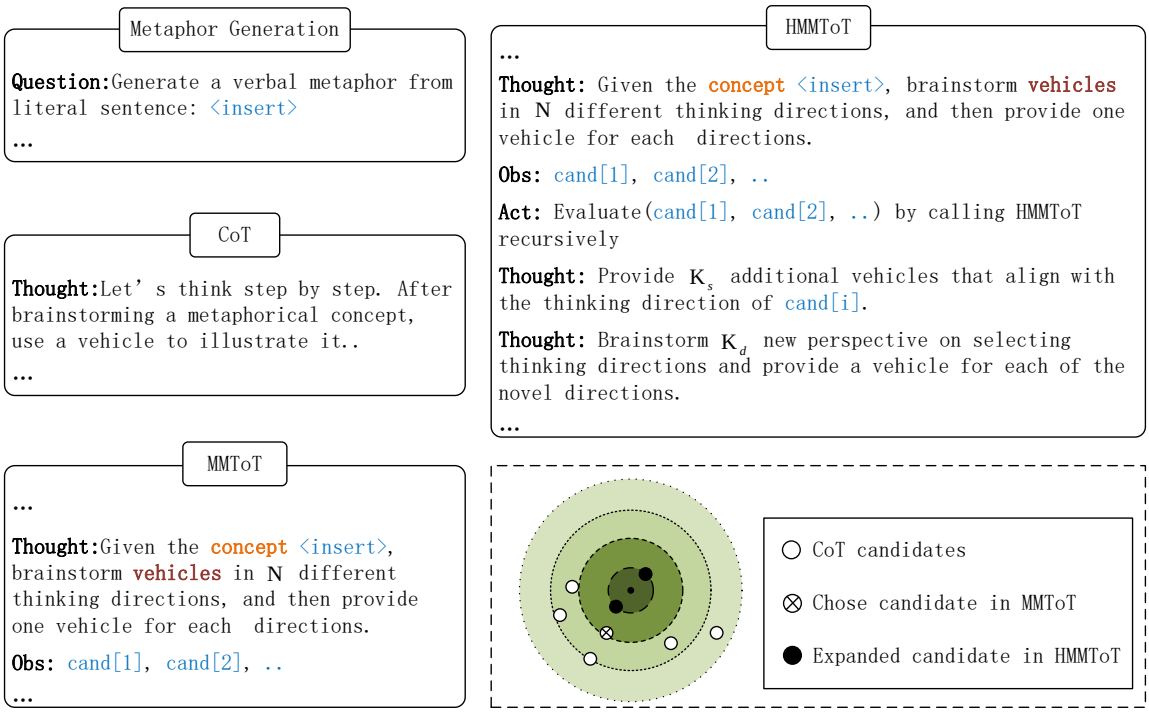


Figure 1: Comparison of 3 prompting methods. Compared with CoT, MMToT and HMMToT have multi-leveled thoughts ("concepts" and "vehicles" in the prompt pieces). Additionally, HMMToT can adjust the exploration based on evaluation feedback. Hence HMMToT may find the best candidates.

generate metaphors with process explanations, aiding individuals in post-processing the metaphor for creative writing. Source code will be released in https://github.com/SunShCh/dynameta.

The main contributions of our work are as follows:

- We channeling the core principles of CMT into a multi-leveled planning approach (MMToT) for metaphor generation, which enriches the theoretical foundations of metaphor generation.
- We propose a heuristic planning method (HMMToT) for structured language generation, which increases the interpretability and control in metaphor generation.
- We demonstrate that thinking abstract-to- embodiment is effective in metaphor generation, which may inspire other generative tasks to exploit multi-leveled approaches.

## 2 Related Works

### 2.1 Metaphor Generation

Previous efforts in metaphor processing have primarily focused on identification (Zhang and Liu, 2022; Li et al., 2023) and interpretation (Wan et al., 2021; Rosen, 2018). In the realm of metaphor generation, Yu and Wan (Yu and Wan, 2019) employ unsupervised methods to extract metaphorically used verbs and their senses from a Wiki corpus. They then train a neural language model for metaphor generation. Brooks and Youssef (Brooks and Youssef, 2020) explore the use of an unsupervised LSTM model to mine text for syntactic patterns that enable the discrimination of creative metaphorical expressions.

Seq2seq models can capture certain aspects of metaphorical language, particularly when trained on a diverse and large dataset that includes examples of metaphors. Stowe et al. (Stowe et al., 2020) focus on building a seq2seq model with a special mask token to convert metaphorical verbs in input text into original metaphorical sentences as output. On the other hand, Chakrabarty et al. (Chakrabarty et al., 2021) focus on maintaining the same meaning through parallel data creation.

In general, BART and T5 achieve more precise results in controlled metaphor generation through pretraining and fine-tuning. However, these models typically require a large amount of training

data to perform well, which can limit their applicability in certain domains and contexts. The utilization of relational knowledge in FrameNet (of California Berkeley, 2006) is highlighted in (Stowe et al., 2021b) to collect data pairs representing mappings between domains. In (Stowe et al., 2021a), knowledge from MetaNet (Dodge et al., 2015) and FrameNet is harnessed to construct metaphorical pairs, demonstrating improved metaphoricity through experimentation. However, the reliance on the inherent concept domains of FrameNet and MetaNet makes it difficult to obtain sufficiently novel metaphor mappings.

### 2.2 LLM Planning for Complex Task Solving

LLM possesses extensive knowledge and language understanding capabilities, enabling it to generate metaphors across various domains, topics, and contexts (Zhao et al., 2023). This significantly enhances its performance in free generation and holds potential for creative metaphor generation. However, LLM struggles with complex task solving, such as commonsense reasoning (Talmor et al., 2019), mathematical reasoning (Qian et al., 2023), and multi-hop question answering (Bian et al., 2024). Current methods often lack control over metaphor generation processes and understanding of the task solving, resulting in somewhat unpredictable and inapplicable outcomes.

Representative works aimed at enhancing LLMs' reasoning abilities include Chain-of-Thought (CoT) (Wei et al., 2022; Shao et al., 2024). A more refined approach suggests using prompt-based planning to decompose intricate tasks into manageable subtasks, thereby creating a strategic action plan to achieve the goal. LLMs excel at devising plans and receiving feedback in the form of natural language. In plan generation, Selfplanning (Jiang et al., 2023) and DECOMP (Khot et al., 2023) incorporate demonstrations in the prompt to guide the LLM's plan formulation. As for feedback acquisition, Personas (Wang et al., 2023) use multiple fine-grained personas in LLMs to enhance problem-solving abilities. Tree of Thoughts (Yao et al., 2023) proposes comparing different plans through voting. By considering multiple reasoning paths and feedback evaluations, ToT employs backtracking to enable global thought choices. We believe that if LLM planning can be applied to metaphor generation, it can better address the issues of control and understanding in LLM's metaphor generation.

## 3 Method

We approach metaphor generation as a planning problem and modify the depth-first planning algorithm to account for the vast thought space involved in creative writing. To provide clarity and reference, we include Table 1, which presents a symbol table containing the notations and their corresponding definitions used in this study.

| Symbol | Description |
|---|---|
| x | literal text |
| z | thought |
| s | state $s = [x; z_1..z_i]$ |
| t | predefined thread of thoughts |
| C | maximum score in evaluation |
| N | initial breadth |
| K | expanding breadth |

Table 1: Symbol Table

### 3.1 Representation of the Generation Procedure

The metaphor generation procedure is represented as a tree that begins with a literal sentence and expands using thoughts as intermediate steps. By utilizing a generator to create new nodes and an evaluator to score the nodes, the planning of the generation procedure can be implemented as a search algorithm. Both Generator and Evaluator can be implemented by prompting a LLM, which is denoted as $\theta$.

#### 3.1.1 State

A state $s$ is a sequence of thoughts $z_1..z_i$ appended to the literal text $x$. The initial state $s_0 = [x]$. When the target metaphor $z^*$ is included in the state list, the generation procedure reaches a goal state $s^* = [x; z_1..z^*]$.

#### 3.1.2 Generator

To achieve the goal state, each thought is sampled sequentially, $p_\theta(z) = \prod_{i=1}^{n} p_\theta(z_i|x, z_{1..(i-1)})$. Practically, we use prompt to augment the thought generation, thus the LLM is denoted as $p_\theta^{prompt}$. In designing the process of metaphor generation, successive stages are predefined as a thread of thoughts, denoted as $t = [z_1..z^*]$. For verbal metaphor, we choose $t = ['concepts', 'vehicle', 'verb', 'metaphor']$. For example, the prompt relating to $'verb'$ could be: *Given the vehicle, list $N$ candidate verbs which*

*draw on tangible experiences and bodily interactions with the world*. Note that we use verb metaphor for example, other type of metaphor can be prompted with a slightly different $t$.

For each non-leaf node, we prompt LLM to generate $N$ candidates. With a rich thought space, the generator samples from an $i.i.d.$ distribution: $z^{(j)} \sim p_\theta(z_{i+1}|s) = p_\theta(z_{i+1}|x, z_{1..i}), j = 1..N$.

#### 3.1.3 State Evaluator

When the leaf node is reached in exploration, metaphor $z^*$ is scored using a state evaluator. To align with user needs, we integrate human criteria into the thread of thoughts $t$ to evaluate both the metaphor and the intermediate thoughts.

1) Fluency (Flu): "To what extent is the generated discourse fluent and easily comprehensible, taking into account the use of suitable word combinations for clarity?"

2) Meaning (Mea): "Do the input and output refer to or imply the same thing?"

3) Creativity (Crea): "How creative and uncommon are the generated metaphors?"

4) Metaphoricity (Meta): "The intensity ranging from strong metaphors that are almost at the core of expression, to moderate metaphors, slight metaphors that require more context to understand, metaphorical hints, and finally, no metaphor."

### 3.2 Heuristics

Due to the inherent diversity in language expression, the breath of the planning problem is incredibly vast. However, before evaluation, LLM can only sample a finite number of candidates, denoted as $N$. There is no guarantee that these candidates include sufficiently effective intermediate thoughts. To address this challenge, we propose a heuristic method for breadth expansion in the planning of metaphor generation.

Intuitively, undiscovered gold deposits are more likely to exist in areas surrounding where gold has already been discovered or in areas that have been overlooked. As in Figure 2, for a given node, we first conduct a depth-first search within a specified breadth. We then identify two types of areas within the node's thought space based on the evaluation results: the area around the best candidate and the overlooked low-density area. We proceed to explore these areas proportionally. This way, our Heuristic Planning of Thoughts (HMMToT) method can adjust both depth and breadth directions to align with user needs, which are expressed as human criteria and integrated into the thoughts thread.

We refer to these two strategies as 'similar' and 'different', respectively. To adjust the ratio between the two strategies, we use a simple yet effective weighting method as in equation 1, where $W_s$ denotes the weight for 'similar' strategy and $W_d$ for 'different' strategy.

$$\begin{aligned} W_s &= \frac{\max(cand)}{\max(cand) + C - \text{average}(cand)} \\ W_d &= \frac{C - \text{average}(cand)}{\max(cand) + C - \text{average}(cand)} \end{aligned} \tag{1}$$

**Algorithm 1:** HMMToT(p, s, K)

**Data:** Input parameters $p$, $s$, and $K$
**Result:** Output value $v$

1 **if** leaf node **then**
2 $\quad v$ = Evaluator($metaphor$);
3 $\quad$ **return** $v$;
4 **end**
5 **else**
6 $\quad cand$ = Generate $N$ candidates using prompt $s$. (see section 3.1.2);
7 $\quad$ **for** $c$ in $cand$ **do**
8 $\quad\quad v$ = HMMToT($p$, $s.append(c)$, $K$);
9 $\quad\quad V$.append($v$);
10 $\quad$ **end**
11 $\quad j$ = argmax($V$);
12 $\quad$ caculate $W_s, W_d$ as in equation 1;
13 $\quad G_s$ = Generate $K_s$ candidates using prompt as in Figure 1;
14 $\quad G_d$ = Generate $K_d$ candidates using prompt as in Figure 1;
15 $\quad new_cand = G_s \cup G_d$ ;
16 $\quad$ **for** $c$ in $new_cand$ **do**
17 $\quad\quad v$ = HMMToT($p$, $s.append(c)$, $K$);
18 $\quad\quad V'$.append($v$);
19 $\quad$ **end**
20 **end**

### 3.3 Planning Algorithm

We now present our HMMToT algorithm, which is outlined in Algorithm 1. The HMMToT algorithm explores generative states recursively. It completes a metaphor generation when it reaches a leaf node and returns an evaluation (lines 1-4). For non-leaf nodes, candidates are generated and the algorithm traverses to the next-level nodes through recursive procedures (lines 5-10), shown as ① in Figure 2.

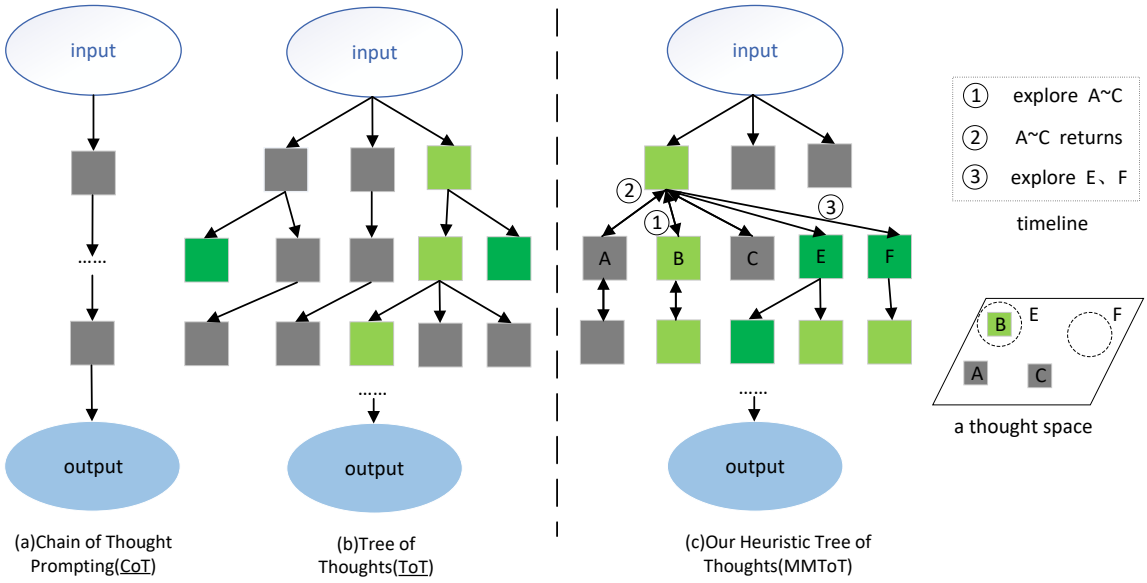


Figure 2: Comparative illustration of various problem solving approaches with LLMs. Each box represents a thought, with dark green and light green colors indicating excellent and relatively good thoughts, respectively. The diamond shape represents the thought space, where region E represents the vicinity of candidate B, and region F represents low density, unexplored space. Our HMMToT involves heuristically expanding the breadth of exploration, as shown in the timeline. Refer to section 3.3 for more details on how breadth expansion is achieved using thought space.

Once the evaluations are received, the algorithm carries out breadth expansion using heuristics (lines 11-15), shown as ② in Figure 2. Then new candidates are explored (lines 16-19), shown as ③ in Figure 2.

**Implementation Details** We initially set the breadth to $N = 4$ and choose an expanding breadth of $K = 4$, considering both averaged scores and token costs. During a 15-sentence experiment, we observed that increasing the value of $K$ resulted in better results. However, we noticed that the rate of improvement decreases after $K = 4$. Moreover, as the breadth increases, the associated token exploration also rises rapidly. Hence, we decided to set the parameter $K$ to 4.

## 4 Experiments

### 4.1 Setup

**Baseline** To compare the quality of generated metaphors, we selected human performance, two well-performing systems, and two methods utilizing GPT-3.5 as baselines. The methods using GPT-3.5 include standard IO prompts and Chain of Thought (CoT) prompts. Despite leveraging literary corpora and knowledge bases for supervised learning, MERMAID and CM-T5, which employ seq2seq models, exhibit weaker pretraining capabilities compared to GPT-3.5, which benefits from a larger dataset. Therefore, for the sake of fairness, IO, CoT, and HMMToT utilized zero-shot methods.

Human: An English professor who is not the author of the paper.

MERMAID: We utilize the model proposed by Chakrabarty et al. (Chakrabarty et al., 2021), where BART is fine-tuned with a parallel corpus. The literal sentences are derived by replacing the metaphorical verbs in the Gutenberg Poetry Corpus with a masked language model.

CM-T5: We use the Controlled Generation model proposed by Stowe etc. (Stowe et al., 2021a), where the source and target domain frames are extracted from MetaNet and Framenet to prompt the T5 model.

IO: Directly using GPT-3.5 to transform literal input $x$ into output $y$.

CoT: Introduces a chain of thoughts with GPT-3.5 to connect $x$ and $y$, where each thought serves as a meaningful intermediate step in problem-solving.

MMToT: The first attempt to construct the ToT metaphor generation. We utilize CMT to break down the generation process into thought steps, and

incorporate human criteria into the state evaluator.

## 4.2 Evaluation Criteria

### 4.2.1 Automatic evaluation

In contrast to other text generation tasks, the auto-evaluation of metaphor generation is challenging due to the rich meanings and emotions inherent in metaphors. While BLEU is commonly used for machine translation evaluation, it relies on word overlap and overlooks semantic similarity, especially when assessing free generation. Thus, we differentiate between BLEU scores for free and controlled generation in GPT-based methods. In controlled generation, metaphors are created by changing only the verbs. Conversely, BERTScore leverages BERT to extract features from sentence pairs and compute similarity matrices, making it a more robust evaluation method. Therefore, we utilize BERTScore as a metric for semantic distance.

### 4.2.2 Human evaluation

The generated metaphors are evaluated by humans, which is a common way to evaluate the generation quality. We engage 10 graduate students majoring in English language. Each graduate student is given a literal input and 6 metaphorical outputs (5 from the baselines and our proposed system, 1 from a human) in a randomized order to avoid potential biases. To reduce the subjectivity of human evaluation, a warm-up step consisting of 10 tests is added before the evaluation begins. After all evaluators complete the same 10 tests, they receive the gold standard scores to help them calibrate their scoring strictness.

To be compatible with previous research (Chakrabarty et al., 2021), the evaluators are instructed to assess the quality of the metaphorical sentences for the given literal sentences across the dimensions as in State Evaluator in section 3.1.

## 4.3 Results and Analysis

To evaluate the model's performance, we randomly chose 150 literal sentences from the data in (Chakrabarty et al., 2021) for assessment. In total, we assessed 900 metaphorical sentences generated by 6 systems.

Results for automatic evaluation on the 150 metaphors are shown in Table 2. As mentioned, we differetiate free and controlled generation in GPT-based methods. MERMAID is better than the other baselines according to BLEU-2(controlled) and BERTScore. In terms of BLEU-2(free), MERMAID is second best.

| System | BLEU-2 (free) | BLEU-2 (controlled) | BERTScore |
|---|---|---|---|
| MERMAID | - | 69 | 0.74 |
| CM-T5 | - | 67 | 0.68 |
| IO | 58 | 66 | 0.71 |
| CoT | 61 | 68 | 0.72 |
| MMToT | **63** | 71 | 0.75 |
| HMMToT | 61 | **72** | **0.77** |

Table 2: Automatic evaluation results

The human evaluation results for metaphor generation methods and human performance on the test set are presented in Table 3.

| System | Flu | Mea | Crea | Meta |
|---|---|---|---|---|
| Human | **4.2** | **4.5** | 3.6 | 4.0 |
| MERMAID | 4.1 | 4.2 | 4.0 | 3.9 |
| CM-T5 | 3.9 | 3.5 | 4.1 | 3.8 |
| IO | 3.8 | 3.6 | 3.5 | 3.5 |
| CoT | 3.7 | 3.9 | 3.6 | 3.7 |
| MMToT | 3.9 | 4.1 | 3.8 | 3.9 |
| HMMToT | 4.1 | 4.3 | **4.2** | **4.1** |

Table 3: Averaged performance of human evaluation

In terms of Fluency and Meaning, Human achieved the highest scores due to the metaphors provided being easier to understand and closer to the literal meaning in Human Evaluation. The lower score of CM-T5 in Meaning is attributed to the inadequacy of domain mapping in the knowledge base when dealing with literal sentences. GPT-3.5 offers extensive domain mapping options, but direct outputs from GPT-3.5 using the IO method are grammatically correct yet less comprehensible. MMToT and HMMToT outperformed CoT in all criteria. By effectively utilizing the domain mapping capabilities within GPT through backtrack in planning, HMMToT demonstrates notable improvements. This not only makes the sentences easier to understand, but also brings them closer to the original meaning. As a result, it surpasses the baselines and approaches human-level performance in terms of Fluency and Meaning.

When assessing creativity and metaphorocity, the evaluation criteria center around originality and the strength of the metaphors. By incorporating supervised learning from literary collections and knowledge databases, MERMAID and CM-T5 perform better in both creativity and metaphorocity

| Literal | System | Metaphor | Flu | Mea | Crea | Meta |
|---|---|---|---|---|---|---|
| The scream filled the night | Human | The scream covered the night | 3.7 | **4.7** | 3.1 | 3.9 |
| | MERMAID | The scream pierced the night | **4.4** | 3.6 | 3.6 | **4.3** |
| | CM-T5 | The scream hounded the night | 3.6 | 2.9 | 4.5 | 4.2 |
| | IO | The scream painted the night with its echoes | 3.6 | 2.7 | **4.6** | 2.8 |
| | CoT | The scream raged through the peaceful night | 3.0 | 3.2 | 4.3 | 3.8 |
| | MMToT | The scream reverberated through the night | 3.5 | 3.8 | 4.2 | 3.9 |
| | HMMToT | The scream infiltrated the night | 4.0 | 4.5 | 4.5 | **4.3** |
| The wildfire spread through the forest at an amazing speed | Human | The wildfire ravaged through the forest at an amazing speed | 4.0 | 4.0 | 4.1 | 3.8 |
| | MERMAID | The wildfire danced through the forest at an amazing speed | 3.9 | 3.3 | **4.5** | 4.1 |
| | CM-T5 | The wildfire ruled through the forest at an amazing speed | 3.5 | 2.8 | **4.5** | 3.6 |
| | IO | The wildfire raced through the forest at an amazing speed | 4.1 | 4.1 | 3.7 | 3.9 |
| | CoT | The wildfire devoured the forest at an amazing speed | **4.4** | 4.3 | **4.5** | **4.2** |
| | MMToT | The wildfire engulfed the forest at an amazing speed | **4.4** | **4.5** | 4.3 | 4.1 |
| | HMMToT | The wildfire consumed the forest at an amazing speed | 4.3 | **4.5** | 4.4 | **4.2** |
| My heart beats when he walks in the room | Human | My heart skips when he walks in the room | **4.6** | **4.8** | 4.0 | 4.2 |
| | MERMAID | My heart jumps when he walks in the room | 4.4 | 4.7 | 4.2 | 4.0 |
| | CM-T5 | My heart criticizes when he walks in the room | 3.7 | 2.8 | 4.2 | 4.2 |
| | IO | My heart dances when he walks in the room | 4.2 | 4.5 | 4.2 | 4.1 |
| | CoT | My heart pounds like a drum when he walks in the room | 4.4 | 4.7 | 4.1 | 4.2 |
| | MMToT | My heart surges when he walks in the room | 4.4 | 4.5 | 4.2 | **4.3** |
| | HMMToT | My heart takes off when he walks in the room | 4.5 | 4.6 | **4.3** | **4.3** |
| After a glass of wine, he relaxed up a bit | Human | After a glass of wine, he uncoiled up a bit | 4.1 | 4.2 | 2.7 | 4.0 |
| | MERMAID | After a glass of wine, he loosened up a bit | 4.2 | 4.5 | **4.6** | 3.8 |
| | CM-T5 | After a glass of wine, he warmed up a bit | **4.3** | 4.2 | 4.3 | 3.8 |
| | IO | After a glass of wine, he mellowed out like a soft melody | 4.0 | 4.0 | 2.8 | **4.3** |
| | CoT | After a glass of wine, he unwound a bit | 4.1 | 4.4 | 2.9 | 4.0 |
| | MMToT | After a glass of wine, he untangled a bit | 4.1 | 4.3 | 4.2 | 4.1 |
| | HMMToT | After a glass of wine, he soared into the skies of relaxation | 4.1 | **4.6** | **4.6** | 4.2 |
| The tax cut will help the economy | Human | The tax cut will fertilize the economy | 4.2 | 4.1 | 4.2 | 3.8 |
| | MERMAID | The tax cut will stimulate the economy | **4.6** | 4.5 | 4.0 | 3.9 |
| | CM-T5 | The tax cut will improve the economy | 4.4 | 4.5 | 3.5 | 3.6 |
| | IO | The tax cut will preserve the financial health of the economy | 3.9 | 3.8 | 2.9 | 3.1 |
| | CoT | The tax cut will nourish the economic prosperity | 3.8 | 3.9 | 3.7 | 3.4 |
| | MMToT | The tax cut will turbocharge the economy | 4.0 | 4.1 | 4.3 | 4.1 |
| | HMMToT | The tax cut will pave a smooth path for the economy | 4.4 | **4.6** | **4.5** | **4.2** |

Table 4: Examples of metaphor generation with evaluations

compared to IO and CoT, which rely on GPT-3.5. HMMToT's improvement comes from expanding thought spaces in intermediate stages, allowing for the exploration of more unique and impactful ideas and source domains. Consequently, HMMToT, also built upon GPT-3.5, outperforms all benchmarks, including human performance.

Table 4 displays a collection of metaphors generated by various systems, accompanied by human assessments based on specific criteria. We notice that the GPT-based methods sometimes make local modifications to the literal sentences to improve expression. For example, in the first literal sentence, the outputs from the IO and CoT methods include phrases like 'with its echoes' and 'peaceful'. While such modifications can enhance the understanding of metaphors, they do not strictly adhere to the standards of verbal metaphors.

From table 4, we can observe that HMMToT consistently achieves higher scores and demonstrates stable performance. This can be attributed to its heuristic planning with dynamic breadth, which enhances both the quality and comprehensibility of metaphor generation. To illustrate this, let's consider the last literal sentence in table 4. The initial sentence generated from the ideas was *The tax cut will revive the economy*, with ratings of (4.4, 3.9, 4.1, 4.1). The lower score was due to the lack of strong candidates in the ideas. However, by employing the Generator($candidates$, 'different', $K_d$) method, a new idea 'Navigating Smooth Waters' was introduced through heuristic breadth expansion. Subsequent steps identified the Source Domain as Geography and the Candidate Verb as Paves. The final sentence generated was: *The tax cut will pave a smooth path for the economy*, achieving ratings of (4.4, 4.6, 4.5, 4.2). This resulted in an overall improvement of 1.2 points compared to the original evaluation. Additionally, the algorithm output provides an explanation: *Best candidate 'pave' from expanded breadth is chosen among 'revive', 'stimulate', etc.*, which can help users understand the generation process.

LLMs can use correlations to solve problems for which humans, in contrast, seem to apply compressed concepts that reflect their real-world experiences (Mitchell and Krakauer, 2023). Can LLM achieve a comparable level of metaphor without using an explicit representation of the concept space? Our experimental results indicate that having explicit control over the process of mapping concepts leads to enhanced performance, thereby highlighting the disparity between LLM's correlation-based approach and conceptualization methods.

**Ablation Study** To further evaluate the effectiveness of the vicinity heuristic and low density heuristic in our proposed method, we conducted ablation experiments using 15 randomly selected literal sentences. The results are presented in table 5, which clearly demonstrates the positive impact of heuristic planning on all metrics. When heuristic planning was removed, the evaluation scores for the four criteria decreased by an average of 0.2 and 0.225 points.

| | Flu | Mea | Crea | Meta |
|---|---|---|---|---|
| w/o vicinity | 4.0 | 4.1 | 4.0 | 3.9 |
| w/o low density | 4.1 | 4.2 | 3.8 | 3.8 |
| HMMToT | 4.2 | 4.3 | 4.2 | 4.1 |

Table 5: Ablation study

## 5 Conclusion

In this paper, we introduce HMMToT, a framework that breaks down the process of metaphor generation into a planning problem and solves it using heuristic search. To tackle the challenge of the vast thought space in LLM generation, especially in creative writing like metaphor generation, our proposed HMMToT utilizes feedback to inspire the exploration of intermediate steps that are more likely to fulfill user requirements. Our method produces superior results and improves the interpretability and controllability of LLM.

One potential application is editing literal sentences in creative writing. We will further investigate purposeful generation to cater to the needs of assistive writing.

## Limitations

Our approach only utilizes the concept space in a tree-form planning, which may not fully exploit the advantages of conceptual metaphor theory. The graph form needs to be further explored.

## References

Ning Bian, Xianpei Han, Le Sun, Hongyu Lin, Yaojie Lu, and Ben He. 2024. Chatgpt is a knowledgeable but inexperienced solver: An investigation of commonsense problem in large language models. In *Proceedings of the 2024 Joint International Conference on Computational Lin-*

*guistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 3098–3110.

Max Black. 1962. *Models and Metaphors: Studies in Language and Philosophy*. Cornell University Press.

Jennifer Brooks and Abdou Youssef. 2020. Discriminative pattern mining for natural language metaphor generation. In *IEEE International Conference on Big Data*, pages 4276–4283.

Tuhin Chakrabarty, Xurui Zhang, Smaranda Muresan, and Nanyun Peng. 2021. MERMAID: metaphor generation with symbolism and discriminative decoding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 4250–4261.

Ellen Dodge, Jisup Hong, and Elise Stickles. 2015. MetaNet: Deep semantic automatic metaphor analysis. In *Proceedings of the Third Workshop on Metaphor in NLP*, pages 40–49.

Xue Jiang, Yihong Dong, Lecheng Wang, Qiwei Shang, and Ge Li. 2023. Self-planning code generation with large language model. *CoRR*, abs/2303.06689.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed prompting: A modular approach for solving complex tasks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Zoltán Kövecses. 2005. *Metaphor in Culture: Universality and Variation*. Cambridge University Press.

George Lakoff and Mark Johnson. 2008. *Metaphors We Live By*. University of Chicago Press.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Yucheng Li, Shun Wang, Chenghua Lin, and Frank Guerin. 2023. Metaphor detection via explicit basic meanings modelling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL, 9-14, 2023*, pages 91–100.

Melanie Mitchell and David C. Krakauer. 2023. The debate over understanding in ai's large language models. *Proceedings of the National Academy of Sciences of the United States of America*, 120.

University of California Berkeley. 2006. The berkeley framenet project.

Andrew Ortony. 1980. Understanding metaphors. Technical Report Technical Report No. 154, University of Illinois.

Ben Prystawski, Paul H. Thibodeau, and Noah D. Goodman. 2023. Psychologically-informed chain-of-thought prompts for metaphor understanding in large language models. In *Proceedings of the 45th Annual Meeting of the Cognitive Science Society, CogSci 2023, Sydney, NSW, Australia, July 26-29, 2023*, pages 2311–2317.

Jing Qian, Hong Wang, Zekun Li, Shiyang Li, and Xifeng Yan. 2023. Limitations of language models in arithmetic and symbolic induction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL, Toronto, Canada, July 9-14, 2023*, pages 9285–9298.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machince Learning Research*, 21:140:1–140:67.

Zachary Rosen. 2018. Computationally constructed concepts: A machine learning approach to metaphor interpretation using usage-based construction grammatical cues. In *Proceedings of the Workshop on Figurative Language Processing, NAACL-HLT, New Orleans, Louisiana*, pages 102–109.

Yujie Shao, Xinrong Yao, Xingwei Qu, Chenghua Lin, Shi Wang, Wenhao Huang, Ge Zhang, and Jie Fu. 2024. CMDAG: A chinese metaphor

dataset with annotated grounds as cot for boosting metaphor generation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING, Torino, Italy*, pages 3357–3366.

Kevin Stowe, Nils Beck, and Iryna Gurevych. 2021a. Exploring metaphoric paraphrase generation. In *Proceedings of the 25th Conference on Computational Natural Language Learning, CoNLL*, pages 323–336.

Kevin Stowe, Tuhin Chakrabarty, Nanyun Peng, Smaranda Muresan, and Iryna Gurevych. 2021b. Metaphor generation with conceptual mappings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP*, pages 6724–6736.

Kevin Stowe, Leonardo F. R. Ribeiro, and Iryna Gurevych. 2020. Metaphoric paraphrase generation. *CoRR*, abs/2002.12854.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, Minneapolis, MN, USA, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Hugo Touvron, Louis Martin, and Kevin Stone etc. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Hai Wan, Jinxia Lin, Jianfeng Du, Dawei Shen, and Manrong Zhang. 2021. Enhancing metaphor detection by gloss-based interpretations. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP*, pages 1971–1981.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2023. Unleashing cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. *CoRR*, abs/2307.05300.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, pages 24824–24837.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, pages 11809–11822.

Zhiwei Yu and Xiaojun Wan. 2019. How to avoid sentences spelling boring? towards a neural approach to unsupervised metaphor generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, Minneapolis, MN, USA, Volume 1 (Long and Short Papers)*, pages 861–871.

Shenglong Zhang and Ying Liu. 2022. Metaphor detection via linguistics enhanced siamese network. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING, Gyeongju, Republic of Korea*, pages 4149–4159.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.