# Metaphor Generation with Conceptual Mappings

**Kevin Stowe**[1], **Tuhin Chakrabarty**[2], **Nanyun Peng**[3]
**Smaranda Muresan**[2], **Iryna Gurevych**[1]
[1]Ubiquitous Knowledge Processing Lab, Technical University of Darmstadt
https://www.informatik.tu-darmstadt.de/ukp/
[2]Columbia University, {tuhin.chakr,smara}@cs.columbia.edu
[3]University of California Los Angeles, violetpeng@cs.ucla.edu

## Abstract

Generating metaphors is a difficult task as it requires understanding nuanced relationships between abstract concepts. In this paper, we aim to generate a metaphoric sentence given a literal expression by replacing relevant verbs. Guided by conceptual metaphor theory, we propose to control the generation process by encoding conceptual mappings between cognitive domains to generate meaningful metaphoric expressions. To achieve this, we develop two methods: 1) using FrameNet-based embeddings to learn mappings between domains and applying them at the lexical level (CM-Lex), and 2) deriving source/target pairs to train a controlled seq-to-seq generation model (CM-BART). We assess our methods through automatic and human evaluation for basic metaphoricity and conceptual metaphor presence. We show that the unsupervised CM-Lex model is competitive with recent deep learning metaphor generation systems, and CM-BART outperforms all other models both in automatic and human evaluations.[1]

## 1 Introduction

Recent neural models have led to important progress in natural language generation (NLG) tasks. While pre-trained models have facilitated advances in many areas of generation, the field of metaphor generation remains relatively unexplored. Moreover, the few existing deep learning models for metaphor generation (Yu and Wan, 2019; Stowe et al., 2020; Chakrabarty et al., 2020) lack any conceptualization of the meaning of the metaphors.

This work proposes the first step towards metaphor generation informed by the conceptual metaphor theory (CMT) (Lakoff and Johnson, 1980; Lakoff, 1993; Reddy, 1979). CMT holds that we use conceptual mappings between domains (conceptual structures that group related concepts) to generate linguistic metaphors.[2] Metaphoric mappings consist of a *source* and a *target* conceptual domain. The *source* domain is the conceptual domain from which we draw the *metaphorical* expressions, while the *target* domain is the conceptual domain that we try to understand. A classical mapping is ARGUMENT IS WAR, in which we conceptualize the *target* argumentation domain as the more concrete *source* domain of war:

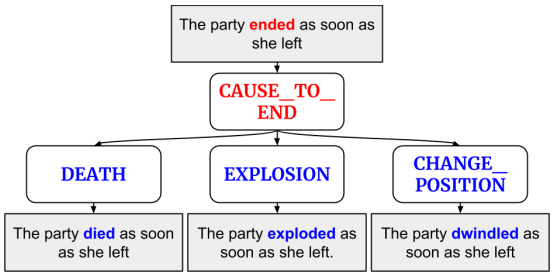


Figure 1: Metaphor generation guided by conceptual metaphors. Given a literal input, we can generate metaphoric outputs based on different mappings between conceptual domains.

- They **fought** against the contract.
- They **defended** their new proposal.

We focus on verbs, as they are often the key component of metaphoric expressions (Steen et al., 2010; Martin, 2006). When used metaphorically, verbs typically evoke source domains (e.g. **fought**, **defended** in the above examples): they are concrete, and are used to understand more abstract targets (i.e., argumentation verbs such as *argued*, *supported*) via conceptual mappings (Sullivan, 2013).

We propose a novel framework for metaphor generation informed by conceptual metaphor theory. Given a *literal* input sentence that evokes a *target domain* we generate *metaphoric* sentences that

[1]All code, models, and data are made available at: https://github.com/UKPLab/acl2021-metaphor-generation-conceptual

[2]"Domains" are also often referred to as "image schema", "frames", "scenes", and more; see Kövecses (2020)

evoke desired corresponding *source* domain(s).[3] For example, given the literal sentence *The party ended as soon as she left* evoking the target domain **CAUSE_TO_END**, we can apply a variety of conceptual mappings to generate different metaphoric outputs evoking different source domains (see Figure 1). This allows us to generate metaphoric expressions that match known metaphoric mappings, as well as generating from unseen mappings to explore novel metaphors. Our contributions are:

- Two metaphor generation models grounded in CMT: 1) An unsupervised lexical model relying on frame embeddings learned from Framenet (CM-Lex, Section 3.1) and 2) a BART (Lewis et al., 2020) model encoding source/target domain information through fine-tuning (CM-BART, Section 3.2).
- Two metaphor generation tasks: 1) generate metaphoric expressions from known concept mappings, for which we provide gold standard test data, and 2) generate novel expressions from unknown metaphors using rare and unseen mappings (Section 4).
- A thorough evaluation using both automatic and human evaluations (Section 5). We show that our CM-BART model improves over all others in terms of metaphoricity (by $\geq 7\%$) and domain evocation (by $\geq 33\%$), and CM-Lex is competitive with previous neural models on metaphoricity while outperforming them on domain evocation (by $\geq 13\%$).

## 2 Task Definition

Traditional metaphor generation models focus only on whether the generated output is in some way "metaphoric" or not. This ignores the semantic and cognitive properties inherent in metaphoricity. These models can, to some degree, generate metaphors given a literal input, but these outputs often do not evoke the intended metaphor.

Controlled metaphor generation yields critical benefits over these uncontrolled systems. For sentences in context, having metaphors that are consistent with the text is essential for natural understanding. Also, metaphors are not only used to express human knowledge, but can also help shape our understanding of the world: having fine-grained control over the generation process allows us to explore novel metaphoric mappings and perhaps improve our understanding of the related domains.

To achieve controlled metaphor generation, we define our task as follows: given *a literal input* sentence which evokes *a target domain* and *an intended conceptual mapping*, generate a metaphoric sentence such that it evokes a desired source domain. Thus, our generation models receive three inputs: 1) a literal input sentence (*They **argued** against the contract*), 2) the target domain evoked by the literal input (**ARGUMENT**) and 3) the desired source domain (**WAR**) for the metaphorical sentence. The output is a metaphorical sentence which evokes the intended mapping (*They **fought** against the contract*)

## 3 Methods

We experiment with two general categories for generation. First, following previous work in metaphor generation and interpretation (Mao et al., 2018; Stowe et al., 2020), we implement lexical methods for replacement, identifying relevant verbs and replacing them with potential candidates for evoking particular mappings. Second, we experiment with deep learning models, employing controlled sequence-to-sequence generation.

### 3.1 CM-Lex

Metaphor generation can be conceptualized as finding key words and replacing them with metaphoric counterparts. This can be done by employing vector spaces, identifying the word most likely to fit in an appropriate context and subjecting them to some constraints of metaphoricity. We build on this paradigm by incorporating facets of conceptual metaphor theory.

Our procedure is as follows: we learn a joint embedded representations for domains and lexical items. We then use the linear transformation between two domains as a mapping, which can be applied to input words from the target domain to generate a word from the source domain. As a proxy for domains, we utilize FrameNet (Baker et al., 1998), which contains semantic frames along with the set of lexical units that evoke them. Frames can be defined as related systems of concepts (Fillmore, 1982), which is exchangeable with the term "domain" used in conceptual metaphor theory (Cruse and Croft, 2004). Thus, we consider the transformation from one frame to another as a proxy for a conceptual metaphoric mapping.

[3] We note that this source and target terminology used here is opposite to that in machine translation.

We first train FrameNet frame embeddings and employ evaluation metrics to ensure their quality. We then apply transformations between domains to literal verbs to generate metaphors grounded in conceptual metaphor theory.

### 3.1.1 Learning Frame Embeddings

In order to exploit FrameNet frames as conceptual domains, we will embed them in vector space. While lexical and contextualized embeddings have proven effective, the field of embedding concepts from lexical resources is less well explored (Sikos and Padó, 2018; Alhoshan et al., 2019). These methods involve tagging raw corpora using automatic FrameNet parsing and then inputting some combination of the original text and the FrameNet information into standard embedding algorithms.

To train and evaluate frame embeddings, we use 211k sentences of Gold annotations used to train the `Open-SESAME` parser (Swayamdipta et al., 2017), along with a variety of other automatically tagged datasets: 250k individual sentence from the Gutenberg Poetry Corpus (Jacobs, 2018), 17k from various fiction section of the Brown Corpus (Francis and Kucera, 1979), and 80k sentences randomly selected from Wikipedia. From this, we extract a 5-word context window for each verb, creating 1.8M verb instances. We then replace the focus verb with its FrameNet frame label (either provided in the Gold data, or tagged via the parser), and train embedding models on the resulting data. This yields joint embedding spaces that contain both common words and FrameNet frame embeddings.

We define two intrinsic metrics to evaluate the quality of our produced embeddings to enable fine-tuning and validation. First, following Sikos and Padó (2018), we can evaluate quality based on the words that evoke that Frame. FrameNet gives a set of lexical units (LUs) that evoke each frame $f$. We calculate the lexical similarity by taking the distance from the mean embedding of "local" words ($w \in f$) to the mean embedding of a random sample $k$ of "distant" words ($w \notin f$):

$$lex(f) = \sum_{w \in f} \frac{cos(E_w, E_f)}{|f|} - \sum_{w \notin f}^{k} \frac{cos(E_w, E_f)}{k}$$

This lexical metric ($lex$) is evaluates whether the frame embedding is similar to words within its frame and dissimilar to those without.

FrameNet also contains linking relations between frames (eg. `used-by`, `uses`), yielding a hierarchy of connected frames. Starting with the assumption that frames connected in the structure should be more similar, we also calculate a structural similarity metric $str$. We follow the same process as above, taking the distance between the mean embedding of the local frames $n \in N$, where $N$ is the immediate neighbors of $f$, to the mean embedding of a sample $k$ of distant frames $n \notin N$.

$$str(f) = \sum_{n \in N} \frac{cos(E_n, E_f)}{|N|} - \sum_{n \notin N}^{k} \frac{cos(E_n, E_f)}{k}$$

We experiment with three lexical embeddings models: word2vec skip-gram (Mikolov et al., 2013), Glove (Pennington et al., 2014), and FastText (Bojanowski et al., 2017). We experiment with 50, 100, and 300 dimensional representations; we find the 50 dimensional word2vec embeddings perform best for both evaluation metrics.[4]

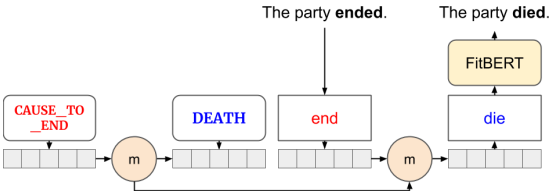

Figure 2: Lexical generation process

### 3.1.2 Embedding Mappings

To apply these embeddings to generate metaphors based on conceptual mappings, we learn mappings between frames and apply the mappings directly to lexical items to facilitate lexical replacement.

We define a mapping $m$ as the pointwise distance between the target frame embedding and the source frame embedding. Following the approach for learning connections between concrete and poetic themes of Gagliano et al. (2016), we sum the embedding of the target verb and the mapping $m$ for the selected conceptual mapping, and select the most similar word to the resulting vector. This word is then delemmatized using `fitbert` (Havens and Stal, 2019) and inserted into the original sentence (Figure 2).

Note that these resulting words are generated without context, as they rely only on the input word and the conceptual mappings. This approach has benefits: we require no labeled metaphor data, using only embeddings trained on FrameNet-tagged corpora. However, ignoring context is likely detrimental. In order to better use contextual information, we explore state-of-the-art sequence-to-sequence modeling.

[4]For full frame embedding evaluation, see Appendix A.

| Literal (filled from LM) | Target Frame | Metaphoric (original) | Source Frame |
|---|---|---|---|
| That tyranny is **destroyed** | DESTRUCTION | That tyranny is **slain** | KILLING |
| The house where love had **ended** | CAUSE_TO_END | The house where love had **died** | DEATH |
| As the moments **passed** on | PROCESS_END | As the moments **roll** on | CAUSE_MOTION |
| What I **learned** my senses fraught | COMING_TO_BELIEVE | What I **bear** my senses fraught | BRINGING |

Table 1: Sample of extracted pairs from the data collection process.

### 3.2 CM-BART

For sequence-to-sequence learning, we fine-tune a pre-trained BART model (Lewis et al., 2020), adding source and target information to guide generation towards the intended metaphors. We first outline a procedure for generating semi-supervised paired data, then detail the training and generation process.

#### 3.2.1 Method for Creating Parallel Data

In order to train sequence-to-sequence models for metaphor generation, we require large scale parallel corpora. We follow the approach of Chakrabarty et al. (2021) and build a corpus of literal/metaphoric paraphrases by starting with the Gutenberg Poetry corpus (Jacobs, 2018), identifying and masking metaphoric verbs, and replacing them with infilling from a language model. We use a BERT-based metaphor classification model trained on the VUA metaphor corpus (Steen et al., 2010) to identify metaphoric verbs in a sentence (i.e "died" in *The house where love had **died***). Then we convert it to a literal sentence *(The house where love had ended)* using infillings from pre-trained BERT (Devlin et al., 2019).

To ensure the literal sentence with replacements convey the same semantic meaning as the metaphorical sentence they are then filtered using symbolic meaning (*SymbolOf* relation) obtained from COMET (Bosselut et al., 2019), a GPT based language model fine-tuned on ConceptNet (Speer et al., 2017). COMET returns top 5 symbolic beams of *(loss, loneliness, despair, sadness and sorrow)* for the sentence "The house where love had **died**" whereas it replaces *sorrow* with *life* for the literal version. While Chakrabarty et al. (2021) filter down to only those candidates with an exact match between the top 5 symbolic beams for the literal and metaphorical sentences returned by the COMET model, we ease the restriction to cases where at least four of five symbols are the same.

In order to learn more direct metaphoric information from this data, we additionally tag each sentence with FrameNet frames using the `Open-SESAME` parser (Swayamdipta et al., 2017). We extract each pair in which both the focus word in the literal, target-domain sentence and the metaphoric, source-domain sentence are assigned a FrameNet frame. We then make the assumption that the relation between the frames for the source and target domains reflects a metaphoric mapping. This then yields a dataset of paired sentences for which we have a metaphoric mapping between domains based on FrameNet for the focus verbs.

Samples of the created data are shown in Table 1. In total this process yields 248k sentences spanning 8.5k unique mappings between FrameNet frames. Each pair comprises a literal and metaphoric sentence, along with the literal target frame and the metaphoric source frame. From these we can directly train a sequence to sequence model for conceptual metaphor-based generation.

#### 3.2.2 Models

We fine-tune BART (Lewis et al., 2020), a pre-trained conditional language model that combines bidirectional and auto-regressive transformers, on the created parallel corpora described in Section 3.2.1. We incorporate representations of the frame information to allow this model to control for the metaphoric mappings evoked.

To transform a literal sentence from a given target domain to a metaphorical sentence evoking a specific source domain, we incorporate both target and source domains (as FrameNet frames) into the textual representation as a control code, following the work of Schiller et al. (2020) who used this procedure for Argument Generation. Following the example from Figure 1, the input literal text fed to the BART encoder would be:

- DEATH $\langle EOT \rangle$ The party $\langle V \rangle$ ended : CAUSE_TO_END $\langle V \rangle$ as soon as she left.

where $\langle EOT \rangle$ and $\langle V \rangle$ are delimiters, DEATH is the source frame, and CAUSE_TO_END the target frame. The decoding target is the metaphoric text "The party **died** as soon as she left", which evokes the CAUSE_TO_END IS DEATH mapping.

Note that our training data differs only at the level of a single verb. We use the generative BART seq2seq model to generate metaphoric paraphrases,

but due to the nature of the training data and the importance of verbs in metaphoric expressions, this is often realized in the output as lexical replacement.

Post fine-tuning, we use top-k (k=5) sampling (Fan et al., 2018) to generate metaphors conditioned on the input literal sentence and source and target domains for the required metaphoric mapping.[5] We evaluate the lexical model (**CM-Lex**) and the sequence-to-sequence model (**CM-BART**) under two experimental settings.

## 4 Experimental Setup

We evaluate our metaphor generation methods against two previous approaches to metaphoric paraphrase generation: the MERMAID system (Chakrabarty et al., 2021) and the metaphor masking model (**MetMask**) (Stowe et al., 2020). We explore two tasks: generating against gold standard metaphoric expressions, and using rare and unseen metaphoric mappings. For the former, we build a gold test set of metaphoric paraphrases that evoke a particular source/target mapping. For the latter, we apply a variety of source/target mappings to literal inputs for which we do not have gold outputs.

### 4.1 Building a Test Set

For a test set, we use the same procedure as our data collection approach from Section 3.2.1. We apply this procedure to two datasets: a sample of the Gutenberg Poetry Corpus and a sample of fiction from the Brown Corpus (Francis and Kucera, 1979). This generates an initial set of literal/metaphoric pairs. We also tagged the pairs from Mohammad et al. (2016) with FrameNet tags, as these generally contain novel, well-formed metaphors. These three datasets each have different properties with regard to metaphor. The **Gutenberg Poetry corpus** has consistent, novel metaphors, but often unconventional syntactic constructions, due to the poetic nature of the text. The **Mohammad 2016** corpus contains manually constructed metaphors which are novel, following relatively basic syntactic patterns. The **Brown Corpus** is standard fiction texts, so the metaphors within tend to be very conventional.

From these sources, we draw pairs randomly, checking that they reflect strong literal/metaphoric paraphrases until we obtain 50 instances from each set. Each pair is tagged with FrameNet frames for the focus verbs, which comprise the metaphoric mapping.[6] For the Brown corpus, metaphoric expressions were relatively rare, and thus valid pairings were sparse: to overcome this, we manually modified 11 of the expressions to evoke the appropriate metaphoric mappings.

In total this process yielded 150 literal/metaphoric pairs, along with the source and target frames that they evoke. We use this dataset to evaluate generating metaphors based on mappings with gold standard outputs, using both automatic and human-based evaluations.

### 4.2 Expanding to Unknown Metaphors

To explore the flexibility of the system developed in this study, we also evaluate them for generation of metaphoric expressions that are not directly linked to gold literal/metaphoric pairs. For this, we begin with our 150 pairs from above, but consider only the literal sentence and the evoked target domain. For each sentence, we generate two source domains that could potentially map to the target. These are selected in order to identify rare and unseen mappings based on the observed mappings in our training data. For **rare** mappings we select a source domain at random from the mappings with the *median* frequency for a given target domain. For **unseen** mappings we select a source domain at random from the FrameNet frames that are *never* used as a source for the given target domain.

This set contains only the tuple (input sentence, target domain, source domain) needed as input to our models; we do not have gold generated metaphorical utterances. Thus, on this set we will only perform human-based evaluation of the quality of the generated metaphors.

### 4.3 Automatic Evaluation Metrics

Word overlap metrics (eg. BLEU, ROUGE) are inherently weak for this task, as these sentences inherently have high overlaps. So instead, we employ semantic distance metrics. We generate sentence embeddings using SBERT[7] (Reimers and Gurevych, 2019) for each of our components: the literal input $L$, the original gold metaphoric expression $M$, and the generated output $G$.

[5] Full parameter tuning outlined in Appendix C.

[6] In 22 cases, parsing errors in FrameNet frames were manually corrected.

[7] Specifically using the `roberta-large` model, which shows the best performance for sentence similarity tasks.

| Model | $dis$ | $rel$ | $mean$ | %= |
|---|---|---|---|---|
| MetMask | .191 | .094 | .143 | .087 |
| MERMAID | .147 | .087 | .117 | .133 |
| CM-Lex | .151 | .086 | .122 | .107 |
| CM-BART | **.085** | **.047** | **.066** | **.293** |

Table 2: Automatic evaluation for metaphor generation systems. **%=** indicates the percentage that matched the gold metaphor exactly.

#### 4.3.1 Distance from Gold Metaphor ($dis$)

The generated metaphoric expressions should match the semantics of the original gold metaphor. We can evaluate this using the cosine distance, here between $M$ and $G$. As SBERT embeddings have been shown to reflect semantic similarity and entailment between paired sentences, this metric should be capable of capturing whether the generated metaphoric expression matches the gold.

#### 4.3.2 Relational distance ($rel$)

Assuming that conceptual metaphoric mappings are responsible for the connecting of meaning between our literal and metaphoric sentences, we would also expect there to be a relation that holds between the original literal input $L$ and metaphoric output $M$. This relation should also hold between the $L$ and the generated metaphor $G$. As a simple metric we can employ cosine distance: we aim for minimizing the distance between $cos(L, M)$ between $cos(L, G)$.

Finally, we include the percentage of times the model produced the exact gold output.

## 5 Results and Analysis

Results for automatic evaluation on the 150 gold metaphors are shown in Table 2. Note that we cannot automatically evaluate against rare or unseen metaphoric mappings, as we lack gold metaphors.

The CM-Lex model is competitive with the best neural baseline, which is encouraging. This shows that simply incorporating basic understanding of conceptual mappings can be a powerful tool for metaphor generation. The CM-BART yields the best automatic performance over all metrics, significantly outperforming all other models ($p < .01$, paired t-test.).

Automatic metrics allow us to quickly prototype metaphoric generation systems based in conceptual metaphor theory. However, they rely on SBERT and inherit the biases and weaknesses therein. We also perform human evaluations, against both the gold test data and the set of rare and unseen mappings.

| | Gold | | Rare | | Unseen | |
|---|---|---|---|---|---|---|
| **Model** | **Met** | **Src** | **Met** | **Src** | **Met** | **Src** |
| MetMask | 2.27 | 1.60 | - | - | - | - |
| MERMAID | 2.56 | 2.12 | - | - | - | - |
| CM-Lex | 2.34 | 2.43 | 2.28 | 2.10 | 1.58 | 1.14 |
| CM-BART | **2.72** | **2.87** | **2.41** | **2.70** | **2.41** | **2.01** |

Table 3: Human evaluations for metaphoricity (**Met**) and source domain evocation (**Src**).

### 5.1 Human Evaluation

For human evaluation, we defined two objectives. First, we aim to capture the **metaphoricity** of the output, as a core objective. The outputs should evoke novel, interesting metaphors regardless of the domains involved. Second, we want the generated metaphoric outputs to evoke the **source** domains (eg. "She **destroyed** his argument" evokes the source domain of WAR).

We recruited three domain experts in metaphoricity. They were instructed to rate each instance on a scale from 1 (not at all) to 4 (very) for metaphoricity and for whether it evokes the source domain. If the sentence was completely unintelligible, they were instructed to mark it as 0 for both categories. For metaphoricity, annotators were given brief definitions of metaphoricity which they incorporated into their expert knowledge to best rate metaphors. For source domain evocation, they were additionally provided with links to the respective FrameNet frames.

We evaluate three different models for the gold metaphors: the best performing previous model, MERMAID, as well as the **lexical** and **CM-BART** models. For all models we evaluate generation using the mappings for our gold test set. For the unknown metaphors without gold sentences, we only evaluate our two controlled models, as the generic baselines give the same output regardless of the intended source. This yields a total of 450 sentences (150 gold, 300 without) that are evaluated for metaphoricity and source domain.

All three experts annotated a random set of 100 training sentences, in order to determine the feasibility and agreement for this task. Agreement rates were .50 for metaphoricity and .37 for source domain (Krippendorff's $\alpha$).[8]

#### 5.1.1 Gold Test Mappings

Results for human evaluations of gold, rare, and unseen metaphoric mappings are shown in Table 3. With regard to the gold mappings, the CM-BART model performs best in metaphoricity and source

[8]Full annotation analysis can be found in Appendix B.

| | Input/TARGET/SOURCE | Model | Output | Met | Src |
|---|---|---|---|---|---|
| 1 | He **resisted** the panic of vertigo<br>SELF_CONTROL IS<br>QUARRELING | Gold | He **fought** the panic of vertigo | | |
| | | MetMask | He got the panic of vertigo | 3 | 1 |
| | | MERMAID | He felt the panic of vertigo | 1 | 2 |
| | | CM-Lex | He confrontations the panic of vertigo | 0 | 0 |
| | | CM-BART | He disputed the panic of vertigo | 3 | 4 |
| 2 | A dim aurora **rises** in my east<br>CHANGE_POSITION_ON_A_SCALE<br>IS RESIDENCE | Gold | A dim aurora **lives** in my east | | |
| | | MetMask | A dim aurora kicked in my east | 3 | 1 |
| | | MERMAID | A dim aurora hangs in my east | 4 | 2 |
| | | CM-Lex | A dim aurora stands in my east | 3 | 3 |
| | | CM-BART | A dim aurora lives in my east | 3 | 4 |
| 3 | People were **running** out of the theater<br>SELF_MOTION IS<br>FLUIDIC_MOTION | Gold | People were **streaming** out of the theater | | |
| | | MetMask | People were clogged out of the theater | 4 | 1 |
| | | MERMAID | People were running out of the theater | 1 | 4 |
| | | CM-Lex | People were boiling out of the theater | 4 | 4 |
| | | CM-BART | People were spilled out of the theater | 4 | 3 |

Table 4: Example outputs of each system along with the mean of their human evaluations.

| | TARGET/SOURCE | Model | Output | Met | Src |
|---|---|---|---|---|---|
| 1 | OPERATE_VEHICLE IS | Input | The car **drove** up alongside him | | |
| | Rare: SELF_MOTION | CM-Lex | The car drove up alongside him | 1 | 1 |
| | | CM-BART | The car ran up alongside him | 4 | 4 |
| | Unseen: DEATH | CM-Lex | The car fell up alongside him | 4 | 4 |
| | | CM-BART | The car died up alongside him | 4 | 2 |
| 2 | DISTRIBUTED_POSITION IS | Input | The meat was **covered** in a fatty gravy | | |
| | Rare: GIVING | CM-Lex | The meat was raised in a fatty gravy | 4 | 1 |
| | | CM-BART | The meat was given in a fatty gravy | 2 | 4 |
| | Unseen: SURRENDERING_POSSESSION | CM-Lex | The meat was cut in a fatty gravy | 1 | 1 |
| | | CM-BART | The meat was yielded in a fatty gravy | 3 | 4 |
| 3 | DISPERSAL IS | Input | At last the darkness began to **dissolve** | | |
| | Rare: ATTEMPT | CM-Lex | At last the darkness began to gorn | 0 | 0 |
| | | CM-BART | At last the darkness began to try | 4 | 4 |
| | Unseen: WARNING | CM-Lex | At last the darkness began to Giffen | 0 | 0 |
| | | CM-BART | At last the darkness began to bite | 4 | 1 |

Table 5: Examples of system outputs on rare and unknown metaphoric mappings.

domain evocation. CM-Lex has middling performance for metaphoricity, but does well at generating correct source domains. The MERMAID system performs well in terms of metaphor generation, but fails to capture the intended source domain.

Examples of each model's generation are shown in Table 4. In 1, we see that CM-Lex generates noise, making the results unintelligible. CM-BART is more robust, generating fluent expressions, and shows evidence of conceptual mapping control, generating a metaphoric expression matching the source domain. In 2, the MetMask and MERMAID models generate reasonable metaphors, which do not evoke the intended domain. CM-Lex is better, generating "stand" which can reflect RESIDENCE, while the CM-BART performs best, generating the gold metaphoric expression.

In 3, we see that the unconstrained models generate effective expressions: "clog" is an evocative metaphor, and "running", while literal, can match the intended domain via the idea of running water. However, our controlled methods both generate novel metaphors that directly evoke the source domain, showing the effectiveness of incorporating conceptual information in generation.

Overall, we see that the unconstrained models often generate good metaphors, but lack consistency with the input, as they are naive with regard to the conceptual backing of these metaphoric expressions. CM-Lex is effective to some degree, even without metaphoric training data, and CM-BART performs best, generating novel metaphors that frequently match the intended metaphoric expression.

### 5.1.2 Unknown Metaphor Mappings

CM-BART outperforms CM-Lex for metaphoricity and source domain evocation for rare and unseen source domains. Examples of the two proposed models' generated for rare and unseen metaphoric mappings are shown in Table 5.

Example 1 shows the ideal case. When given a source domain from a "rare" mapping, the resulting metaphor is fairly reasonable. CM-BART generates a metaphor consistent with the original semantics; CM-Lex generates the literal utterance. When presented with an unseen mapping in which oper-

ating a vehicle is framed as death, we get diverse expressions, both adding meaning to the original utterance. CM-Lex uses the verb "fell" (albeit incorrectly conjugated), which can be used to abstractly evoke the death domain, while CM-BART directly uses the verb "die". The original expression can be ambiguous as to whether the car stopped: the evoked metaphor enforces the stoppage of the car, and also provides color to the expression.

Example 3 highlights a key issue: when the source and target domains are too incongruent, the generated expressions can be inconsistent. CM-Lex here again generates noise. However, CM-BART generates normal, expressive metaphors, which are nonetheless incompatible with the original literal input, which denotes the lessening of darkness. Rather, CM-BART generates a metaphor expressing perhaps growing darkness with the verb **try** and a dangerous darkness with the verb **bite**.

This is a critical point with regard to conceptual mappings. Not all pairs are available: they require semantic consistency, and while generating from any two pairs may yield insightful, interesting, and perhaps inspiring new metaphoric expressions, generating metaphoric paraphrases requires additional knowledge of which source/target pairings are compatible. This generally supports notion of invariance and structure mapping, in which there is inherent structure within domains that needs to be consistent in order to evoke metaphoric mappings between them (Gentner, 1983; Lakoff, 1993).

It must be noted that the systems proposed here have a distinct advantage in this task: we add FrameNet frames, which, while neither perfect nor designed to capture metaphoricity, provide a strong signal for which domains to generate in. This highlights a possible benefit to the interaction between deep, pre-trained models such as BART and available lexical resources: by combining these, we are able to leverage the strength of each to build a powerful metaphor generation system.

## 6 Related Work

We broadly cover two areas of related work: previous computational approaches to CMT, and previous approaches to metaphor generation.

**Computational Approaches to CMT.** There are a variety of approaches to identifying conceptual metaphors themselves. The CorMet system (Mason, 2004) was built to extract conceptual metaphors based on selectional preferences of verbs. Shaikh et al. (2014a) builds "conceptual spaces" for source domains, using rule-based extraction of relations between lexical items. These conceptual spaces are then used to find new conceptual metaphors. This process is extended to build a repository of linguistic and conceptual metaphors (Shaikh et al., 2014b). Mohler et al. (2014) focus on identifying appropriate source domains for metaphoric expressions, using vector-based approaches for metaphor interpretation.

The idea of using frames to represent metaphoric domains has been explored in the MetaNet project (Dodge et al., 2015). We however, restrict our work to FrameNet due to the coverage and availability of reliable automatic parsing.

**Metaphor Generation.** Early work in metaphor generation was based in heuristics, learning to generate relatively simple "A is like B" representations (Abe et al., 2006; Terai and Nakagawa, 2010). In a similar vein, Veale (2016) uses template-like structures to generate creative and metaphoric tweets.

Other works focus on identifying metaphoric mappings using WordNet clustering and selectional preferences (Mason, 2004; Gandy et al., 2013), syntactic relations to build proposition databases (Ovchinnikova et al., 2014), and embedding based approaches to identify poetic relationships (Gagliano et al., 2016). However, the goal of these works is to generate mappings, rather than linguistic expressions that evoke them.

Amongst deep learning approaches Yu and Wan (2019) identify literal and metaphoric words in corpora based on selectional restrictions, and using these to train sequence-to-sequence models for metaphor generation, albeit without reference to any input expression. Stowe et al. (2020) generates metaphors using masked language modeling, masking metaphoric tokens in training in order to encourage metaphoric generation. Other approaches use novel methods for collecting literal/metaphor pairs, training sequence-to-sequence models for simile generation and metaphoric paraphrasing (Chakrabarty et al., 2020, 2021). These approaches effectively generate figurative language, but the models have no knowledge of the underlying metaphors, and thus simply generate ungrounded expressions. This leads to outputs which are possibly metaphoric, but contain no connection to the input, eschewing the critical connections that make novel metaphors powerful. We instead propose methods for generating metaphoric para-

phrases grounded in CMT.

## 7 Conclusions and Future Work

In summary, we have shown two methods for incorporating knowledge of conceptual metaphor theory in metaphor generation. We trained FrameNet frame embeddings to represent conceptual domains, and applied shifts between them to generate metaphors in an unsupervised fashion. Leveraging FrameNet further, we build a dataset of semi-supervised pairs that evoke conceptual metaphors, which can be used along with BART for controlled metaphor generation. This model achieves state-of-the-art performance in metaphor generation by both automatic and human evaluations.

Future work can expand these models to go beyond verbs, incorporating nominal and other types of metaphors. The next necessary step is to go beyond lexicalized metaphors: good, consistent conceptual metaphors often span long stretches of text, and we need to design models that can learn and generate metaphors over larger texts.

## Ethical Considerations

Although we use language models trained on data collected from the Web, which have been shown to have issues with bias and abusive language (Sheng et al., 2019; Wallace et al., 2019), the inductive bias of our models should limit inadvertent negative impacts. Unlike model variants such as GPT, BART is a conditional language model, which provides more control of the generated output. It should also be noted that our CM-BART model is fine-tuned on the poetry corpus which is devoid of harmful and toxic text especially targeted at marginalized communities

Advances in generative AI inherently come with concerns about models' ability to deceive, persuade, and misinform. Metaphorical language has been shown to express and elicit stronger emotion than literal language (Citron and Goldberg, 2014; Mohammad et al., 2016) and to provoke emotional responses in the context of political discourse covered by mainstream newspapers (Figar, 2014). We understand there may be concerns about building generative models for metaphors aimed at persuasion. Social scientists distinguish persuasion from manipulation based on two aspects: dissimulation and constraint (Nettel and Roque, 2012). Dissimulation involves concealing intention, which requires hiding information, whereas constraint involves removing options from the audience and forcing them to accept the conclusion. Our work on metaphor generation does not aim to hide information about a topic or present it as the only choice, but aims to provide the same sentence using more expressive language.

## References


Keiga Abe, Sakamoto Kayo, and Masanori Nakagawa. 2006. A computational model of the metaphor generation process. In *Proceedings of the 28th Annual Meeting of the Cognitive Science Society*, pages 937–942, Vancouver, Canada. Psychology Press.

Waad Alhoshan, Riza Batista-Navarro, and Liping Zhao. 2019. Semantic frame embeddings for detecting relations between software requirements. In *Proceedings of the 13th International Conference on Computational Semantics - Student Papers*, pages 44–51, Gothenburg, Sweden. Association for Computational Linguistics.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Tuhin Chakrabarty, Smaranda Muresan, and Nanyun Peng. 2020. Generating similes effortlessly like a pro: A style transfer approach for simile generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6455–6469, Online. Association for Computational Linguistics.

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*, pages 4250–4261, Online. Association for Computational Linguistics.

Francesca MM Citron and Adele E Goldberg. 2014. Metaphorical sentences are more emotionally engaging than their literal counterparts. *Journal of cognitive neuroscience*, 26(11):2585–2595.

D. Alan Cruse and William Croft. 2004. *Cognitive Linguistics*. Cambridge Textbooks in Linguistics. Cambridge University Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Erik-Lân Do Dinh, Hannah Wieland, and Iryna Gurevych. 2018. Weeding out conventionalized metaphors: A corpus of novel metaphor annotations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1424, Brussels, Belgium. Association for Computational Linguistics.

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, Denver, Colorado. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Vladimir Figar. 2014. Emotional appeal of conceptual metaphors of conflict in the political discourse of daily newspapers. *Facta Universitatis, Linguistics and Literature*, 12(1):43–61.

Charles Fillmore. 1982. Frame Semantics. *Linguistics in the Morning Calm*, 1:111–138.

W. N. Francis and H. Kucera. 1979. Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US.

Andrea Gagliano, Emily Paul, Kyle Booten, and Marti A. Hearst. 2016. Intersecting word vectors to take figurative language to new heights. In *Proceedings of the Fifth Workshop on Computational Linguistics for Literature*, pages 20–31, San Diego, California, USA. Association for Computational Linguistics.

Lisa Gandy, Nadji Allan, Mark Atallah, Ophir Frieder, Newton Howard, Sergey Kanareykin, Moshe Koppel, Mark Last, Yair Neuman, and Shlomo Argamon. 2013. Automatic identification of conceptual metaphors with limited knowledge. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, pages 328–334, Bellevue, Washington. AAAI Press.

Dedre Gentner. 1983. Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science*, 7:1–5.

Sam Havens and Aneta Stal. 2019. Use bert to fill in the blanks.

Arthur M Jacobs. 2018. The Gutenberg English poetry corpus: exemplary quantitative narrative analyses. *Frontiers in Digital Humanities*, 5:5.

Zoltán Kövecses. 2020. *Extended Conceptual Metaphor Theory*. Cambridge University Press.

George Lakoff. 1993. The Contemporary Theory of Metaphor. In Andrew Ortony, editor, *Metaphor and Thought*, pages 202–251. Cambridge University Press.

George Lakoff and Mark Johnson. 1980. *Metaphors We Live By*. University of Chicago Press, Chicago and London.

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, Online. Association for Computational Linguistics.

Rui Mao, Chenghua Lin, and Frank Guerin. 2018. Word embedding and WordNet based metaphor identification and interpretation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1222–1231, Melbourne, Australia. Association for Computational Linguistics.

James H Martin. 2006. A corpus-based analysis of context effects on metaphor comprehension. Technical report.

Zachary J. Mason. 2004. CorMet: A computational, corpus-based conventional metaphor extraction system. *Computational Linguistics*, 30(1):23–44.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Saif Mohammad, Ekaterina Shutova, and Peter Turney. 2016. Metaphor as a medium for emotion: An empirical study. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 23–33, Berlin, Germany. Association for Computational Linguistics.

Michael Mohler, Bryan Rink, David Bracewell, and Marc Tomlinson. 2014. A novel distributional approach to multilingual conceptual metaphor recognition. In *Proceedings of COLING 2014, the 25th*

*International Conference on Computational Linguistics: Technical Papers*, pages 1752–1763, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Ana Laura Nettel and Georges Roque. 2012. Persuasive argumentation versus manipulation. *Argumentation*, 26(1):55–69.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Ekatarina Ovchinnikova, Vladimir Zaytsev, Suzanne Wertheim, and Ross Israel. 2014. Generating conceptual metaphors from proposition stores. cs.CL/1409.7619.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Michael Reddy. 1979. The Conduit Metaphor : A case of frame conflict in our language about language. In Andrew Ortony, editor, *Metaphor and Thought*, pages 284–324. Cambridge University Press, Cambridge.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. 2020. Aspect-controlled neural argument generation. *arXiv preprint arXiv:2005.00084*.

Samira Shaikh, Tomek Strzalkowski, Kit Cho, Ting Liu, George Aaron Broadwell, Laurie Feldman, Sarah Taylor, Boris Yamrom, Ching-Sheng Lin, Ning Sa, Ignacio Cases, Yuliya Peshkova, and Kyle Elliot. 2014a. Discovering conceptual metaphors using source domain spaces. In *Proceedings of the 4th Workshop on Cognitive Aspects of the Lexicon (CogALex)*, pages 210–220, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Samira Shaikh, Tomek Strzalkowski, Ting Liu, George Aaron Broadwell, Boris Yamrom, Sarah Taylor, Laurie Feldman, Kit Cho, Umit Boz, Ignacio Cases, Yuliya Peshkova, and Ching-Sheng Lin. 2014b. A multi-cultural repository of automatically discovered linguistic and conceptual metaphors. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2495–2500, Reykjavik, Iceland. European Language Resources Association (ELRA).

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3407–3412, Hong Kong, China. Association for Computational Linguistics.

Jennifer Sikos and Sebastian Padó. 2018. Using embeddings to compare FrameNet frames across languages. In *Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*, pages 91–101, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *In Thirty-First AAAI Conference on Artificial Intelligence.*, pages 4444–4451, San Francisco, California.

Gerard J. Steen, Aletta G. Dorst, J. Berenike Herrmann, Anna Kaal, Tina Krennmayr, and Trijntje Pasma. 2010. *A Method for Linguistic Metaphor Identification: From MIP to MIPVU*. John Benjamins.

Kevin Stowe, Leonardo Ribeiro, and Iryna Gurevych. 2020. Metaphoric paraphrase generation. *arXiv preprint arXiv:2002.12854*.

Karen Sullivan. 2013. *Frames and Constructions in Metaphoric Language*. John Benjamins.

Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A. Smith. 2017. Frame-Semantic Parsing with Softmax-Margin Segmental RNNs and a Syntactic Scaffold. *arXiv preprint arXiv:1706.09528*.

Asuka Terai and Masanori Nakagawa. 2010. A computational system of metaphor generation with evaluation mechanism. In *International Conference on Artificial Neural Networks*, pages 142–147, Thessaloniki, Greece. Springer.

Tony Veale. 2016. Round up the usual suspects: Knowledge-based metaphor generation. In *Proceedings of the Fourth Workshop on Metaphor in NLP*, pages 34–41, San Diego, California. Association for Computational Linguistics.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.

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*, pages 861–871, Minneapolis, Minnesota. Association for Computational Linguistics.

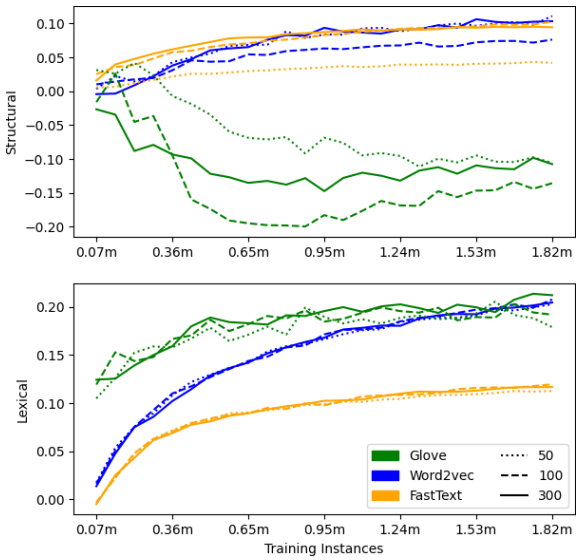


Figure 3: Frame embedding evaluation metrics as data is added.

## A Appendix A

Results for each frame embedding method using the distance metrics defined in Section 3.1 are shown in Table 6.

Figure 3 tracks these evaluation metrics as more data is added to each algorithm. The lexical evaluation relatively stable, peaking in most cases between .1 and .2. The word2vec embeddings maintain their upward progression even at maximal data: theoretically additional data could improve these embeddings further. The structural evaluation shows something very different: while word2vec and FastText embeddings improve as data is added, showing some effects of model size, the Glove embeddings trend sharply negative at first before proceeding beginning to improve.

## B Appendix B

Agreement rates were measured using Krippendorff's $\alpha$. For metaphoricity, the mean score was .505, indicating moderate agreement. However, given the difficulty of this task, we believe this to be relatively stronger: see Table 7 for comparison to other work evaluating metaphor generation.

For source domain annotation, annotators varied in the degree to which source domains were evoked. Initial agreement was relatively poor (.249): we performed a post-processing step, normalizing their results to a consistent mean. This yields an agreement score of .387: which we deemed competitive for the difficulty of the task. As we have no direct comparison for evaluation, further work is required

| Dimensions | *lex* sim | | | *str* sim | | | mean | | |
|---|---|---|---|---|---|---|---|---|---|
| | 50 | 100 | 300 | 50 | 100 | 300 | 50 | 100 | 300 |
| word2vec | .203 | **.208** | .205 | **.111** | .076 | .104 | **.157** | .144 | .154 |
| fasttext | .113 | .120 | .117 | .042 | .103 | .095 | .077 | .111 | .106 |
| glove | .179 | .191 | .212 | -.106 | -.136 | -.108 | .037 | .028 | .052 |

Table 6

| Paper | n | Method | Agreement |
|---|---|---|---|
| Do Dinh et al. (2018) | 15,180 | MTurk | .16-.38 $\alpha$ |
| Yu and Wan (2019) | 80 | MTurk | - |
| Chakrabarty et al. (2020) | 900 | MTurk | .36-.49 $\alpha$ |
| Stowe et al. (2020) | 513 | MTurk | - |
| Chakrabarty et al. (2021) | 900 | MTurk | - |
| This work | 450 | Experts | .505 $\alpha$ |

Table 7: Comparison of agreement rates for various metaphor evaluation tasks. Note that Do Dinh et al. (2018) developed a real-valued scoring layer over an existing corpus rather than evaluating generated outputs. "-" indicates agreement is not reported.

to refine this type of evaluation process.

## C Appendix C

For retrieving commonsense symbolism of the sentences, we use the pre-trained COMET model [9] and retrieve top 5 candidates for each input.

1. **No of Parameters:** We use the BART large checkpoint (400M parameters) and use the FAIRSEQ implementation (Ott et al., 2019) [10].
2. **No of Epochs:** We fine-tune pre-trained BART for 25 epochs for CM-BART model and save the best model based on validation perplexity.
3. **Training Time:** Our training time is 60 minutes for CM-BART.
4. **Hardware Configuration:** We use 4 RTX 2080 GPUs.
5. **Training Hyper parameters:** We use the same parameters as the FAIRSEQ github repository where BART was fine-tuned for the CNN-DM summarization task with the exception of the size of each mini-batch, in terms of the number of tokens, for which we used 1024. [11]
6. **Decoding Strategy & Hyper Parameters:** For decoding we generate metaphors from our models using a top-k random sampling scheme (Fan et al., 2018). At each timestep, the model generates the probability of each word in the vocabulary being the likely next word. We randomly sample from the k = 5 most likely candidates from this distribution.

[9] https://github.com/atcbosselut/comet-commonsense

[10] https://github.com/pytorch/fairseq/tree/master/examples/bart

[11] https://github.com/pytorch/fairseq/blob/master/examples/roberta/README.glue.md