Exploring Metaphoric Paraphrase Generation

Kevin Stowe, Nils Beck, Iryna Gurevych
Ubiquitous Knowledge Processing Lab, Technical University of Darmstadt
https://www.informatik.tu-darmstadt.de/ukp/

Abstract

Metaphor generation is a difficult task, and has seen tremendous improvement with the advent of deep pretrained models. We focus here on the specific task of metaphoric paraphrase generation, in which we provide a literal sentence and generate a metaphoric sentence which paraphrases that input. We compare naive, "free" generation models with those that exploit forms of control over the generation process, adding additional information based on conceptual metaphor theory. We evaluate two methods for generating paired training data, which is then used to train T5 models for free and controlled generation. We use crowdsourcing to evaluate the results, showing that free models tend to generate more fluent paraphrases, while controlled models are better at generating novel metaphors. We then analyze evaluation metrics, showing that different metrics are necessary to capture different aspects of metaphoric paraphrasing. We release our data and models, as well as our annotated results in order to facilitate development of better evaluation metrics.[1]

1 Introduction

Metaphors are ubiquitous in human language, and while humans seem capable of easily understanding even complex metaphors, it remains difficult to implement computational methods that capture the depth and breadth of meaning inherent in novel metaphors. While many approaches have been implemented for metaphor detection, interpretation, and generation, there remain many open questions about how to incorporate linguistic and cognitive theory into these methods, as well as how to evaluate them quickly and effectively.

Our task is that of metaphoric paraphrase generation: given a literal input sentence, we aim to generate a semantically similar sentence that is metaphoric. Specifically, we focus on two possible methods for metaphoric paraphrase generation: **free** and **controlled**.

In **free** generation, models are trained using literal and metaphoric pairs, but are not given information regarding which metaphoric meaning is intended. This allows the model the freedom to develop new metaphors that can be creative, novel, and perhaps even inspiring. However, it can lead to noisy generation, which is difficult to evaluate as even randomly generated metaphors can sometimes be effective via coercion-like processes. Additionally, if a generated text is to remain coherent, metaphor generation must be consistent within it, and free generation is thus ill-suited to generating longer coherent metaphoric texts.

The other option, **controlled** generation, involves adding additional constraints the generation objective to encourage the model to generate a specific metaphor. This methodology is useful when an intended metaphor is known, or needs to be consistent across a dialogue: the model can be suitably constrained to be consistent. Recent work has explored controlled generation (Stowe et al., 2021); we aim to address if and how adding control improves over free generation.

As a theoretical basis for metaphors, we employ Conceptual Metaphor Theory (CMT). In CMT, we consider linguistic metaphors as arising from conceptual metaphors that are a part of our cognitive processes, in which a concrete *source* domain is used to better understand a more abstract *target* domain (Lakoff and Johnson, 1980; Lakoff, 1993). We typically consider language from the abstract target domain in the context sentence as literal, and aim to generate expressions evoking a concrete source domain that can be used to describe it metaphorically. In free generation, the system is left open to choose which domain is appropriate; in controlled generation, we provide the model explicitly with the required domains.

[1] Code and data at https://github.com/UKPLab/conll2021-metaphoric-paraphrase-generation.

Consider the following example of free generation. We train a seq2seq metaphor model to take a literal sentence as the context and produce some metaphoric hypothesis:

1. The company was losing money rapidly. $\Longrightarrow$ The company was leaking money.

Contrast this with controlled generation, in which we have an intended metaphor: we constrain the model to produce an specific metaphor ie. viewing company as a human body, and the loss being some kind injury:

1. (ECONOMIC HARM IS PHYSICAL INJURY) The company was losing money rapidly. $\Longrightarrow$ The company was hemorrhaging money.

We start by implementing the metaphor-naive SOW-REAP paraphrase generation system of Goyal and Durrett (2020), to compare it to those designed specifically for metaphoric generation. We then fine tune pretrained T5 seq2seq models using literal/metaphoric pairs, as they allow for easy implementation of control codes (Raffel et al., 2020). We use both an available dataset of semi-supervised literal/metaphoric pairs and a new dataset based on a novel pair generation procedure. We evaluate the effectiveness of free and controlled generation, using crowdsourcing to label the outputs for fluency, paraphrase quality, and metaphoricity. Using our resulting crowdsourced dataset, we analyze evaluation metrics: we assess whether traditional evaluation metrics for generation are effective for metaphoric language. We explore a suite of possible metrics, assessing their correlation to human judgments across the three crowdsourced labels.

Our contribution is summarized as follows:

- We compare controlled and free metaphor generation, showing that adding control improves the metaphoricity of outputs, while free generation tends to generate more fluent, coherent paraphrases.
- We perform an analysis of automatic evaluation metrics, comparing them to crowdsourced annotations, showing conflict between fluency and metaphoricity evaluation metrics, indicating that individuals metrics are poor evaluators for metaphor generation.
- We release our novel training dataset of 360k pairs based on MetaNet mappings, along with our gold test set and 1,250 samples annotated for fluency, sentence similarity, and metaphoricity to allow for better evaluation of metaphor generation systems.

2 Related Work

2.1 Paraphrase Generation

The task of paraphrase generation has a rich background, including rule-based approaches (McKeown, 1983) and mono-lingual machine translation methods (Quirk et al., 2004; Wubben et al., 2010). Deep learning has driven the field in recent years, particularly autoencoder and LSTM networks (Gupta et al., 2018; Prakash et al., 2016) and transformer-based methods (Li et al., 2019; Egonmwan and Chali, 2019; Wang et al., 2019).

Our task, however, diverges in many regards from the standard task of paraphrasing, which typically relies on syntactic and lexical transformations to generate sentences with exactly the same meaning as the input. We instead aim to generate sentences that contain additional meaning via a metaphoric mappings. In this regard, the entailment relations and semantics of generated metaphors will differ from traditional paraphrase generation: we expect the generated metaphor to entail the literal context, but as it adds something additional, the literal context may not entail the metaphoric output.[2]

Recent work in paraphrase generation has taken a step in this direction by focusing on generating diverse paraphrases (Xu et al., 2018; Qian et al., 2019; Yang et al., 2019; Goyal and Durrett, 2020). While not designed for metaphoricity, we include the SOW-REAP system of Goyal and Durrett (2020) as a comparison to highlight the disparity between metaphoric and non-metaphoric paraphrasing capabilities.

2.2 Metaphor Generation

Early work in computational metaphor generation involves generating simple "A is like B" expressions, based on probabilistic relationships between words (Abe et al., 2006; Terai and Nakagawa, 2010). These methods are effective to a degree, but lack the flexibility necessary to instantiate natural language metaphors.

[2]Consider the metaphor "Her husband abuses alcohol." (Mohammad et al., 2016): it entails the literal paraphrase "Her husband drinks alcohol", but the reverse is not necessarily true.

Metaphor generation has recently seen significant advances due to deep pretrained language models. Yu and Wan (2019) use neural models to generate metaphoric expressions in an unsupervised manner. They identify source and target verbs automatically from corpora, and use these to train a neural language model. However, they are generating metaphors without regard to reference texts from metaphorically trained language models, and the outputs bear no relation to the inputs.

With regard to paraphrasing, Stowe et al. (2020) use a metaphor masking process to generate parallel training data in which key metaphoric words are hidden, causing the resulting seq2seq model to generate metaphoric words. Chakrabarty et al. (2020) build a simile generation system based on pretrained seq2seq models. Similarly, the MERMAID system uses a semi-supervised data collection method to generate metaphoric pairs, using them to fine-tune a BART-based seq2seq model (Chakrabarty et al., 2021). These models are restricted to **free** generation: the models are not constrained to generate in a particular domain or metaphoric mapping. It may be the case with sufficient large pretrained models, free systems learn to generate valid conceptual metaphors.

For controlled generation, Stowe et al. (2021) collect pairs based on FrameNet frame tags, which are used to represent conceptual domains. These are then used to build a controlled paraphrasing system. We undertake the task of comparing these two methodologies: do free generation systems generate valid metaphors, and does adding conceptual metaphor information improve metaphoric paraphrase generation?

3 Data

A key bottleneck in metaphoric paraphrasing is the lack of high quality literal/metaphoric pairs for both training and evaluation. While two datasets of hand-crafted metaphoric paraphrases are available (Mohammad et al., 2016; Bizzoni and Lappin, 2018), they both contain less than 200 instances, and are thus too small for model training and/or fine-tuning. We explore two possible datasets for this purpose: a previously used dataset which contains source and target domains via FrameNet frames (Stowe et al., 2021), and a new semi-supervised dataset which contains source/target information based on the metaphor-based lexical resource MetaNet.

3.1 Source/Target pairs from FrameNet Tagging

As a starting point, we use the data from Stowe et al. (2021). This dataset is a semi-supervised pairing of literal and metaphoric sentences. It is generated by taking sentences from the Gutenberg Poetry corpus (Jacobs, 2018), tagging them automatically with metaphor labels, then replacing metaphoric words using a language model to produce a literal paraphrase. This follows the assumption that the more likely replacement words from the language model will tend to be literal. The literal paraphrase and original metaphor are tagged using the COMET parser (Bosselut et al., 2019), and pairs that contain overlapping common-sense symbols are kept. They then use the open-SESAME parser to add FrameNet frame labels, which function as domains. The pairs are considered to reflect metaphoric mappings between their respective frame labels, which are then provided as control codes to a BART model to generate metaphors from specific mappings (Lewis et al., 2020).

This method is effective for generating controlled metaphor pairs, but has a number of drawbacks. First, it relies heavily on a number of models (metaphor classification, FrameNet frame tagging, and COMET symbol extraction), each of which introduces error. Specifically, metaphor detection remains a difficult task (with state-of-the-art results $<$ .77 F1 (Leong et al., 2020)), making the identification of initial metaphors difficult. FrameNet tagging is also error prone, with the micro-F1 score for frame tagging being 70.9. The data is all extracted from a single source, the Gutenberg Poetry corpus, and thus inherits biases from the data: the sentences are all relatively short, and often syntactically strange due to the poetic style of the corpus.

Finally, the mappings learned are entirely reliant on the FrameNet frame tags: the mappings themselves are still detached from any theory, and may not reflect true metaphoric mappings.To obviate these difficulties, we here propose an alternative method for generating data which uses the MetaNet resource, which instead contains gold metaphoric mappings.

3.2 MetaNet Generation

MetaNet is a resource which contains a substantial set of mappings between conceptual domains (Dodge et al., 2015). These mappings consist of source and target domains which evoke the concep-

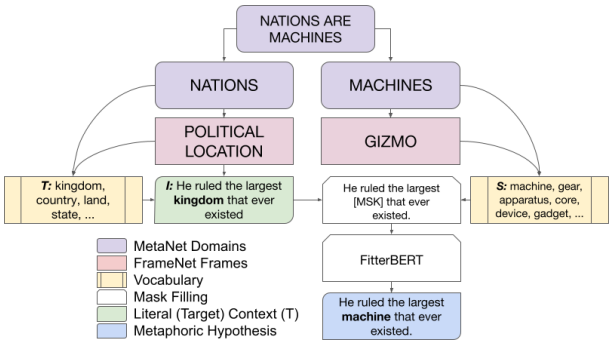

Figure 1: MetaNet pair generation process. By linking MetaNet conceptual mappings to FrameNet frames, we identify input sentences ***I*** from the target domain, mask the target the target words ***T***, and generate hypotheses using a set of candidate words ***S*** from the mapped source domain.

tual metaphor, along with links from these domains to FrameNet frames

For each mapping in MetaNet, we build a set of input sentences I, a set of source vocabulary words S, and a set of target vocabulary words T. We start with these conceptual mappings from MetaNet, which are based in CMT. We extract the target domain (which is typically found in the literal sentence) and the source domain (which we typically consider to be metaphoric) from each mapping. Note that verbs are typically what we consider the metaphoric element of these phrases, and they typically evoke source domains(Deignan, 2005; ste; Sullivan, 2013).

From these mappings, we follow MetaNet's links from the target domains to their respective FrameNet frames. We then expand the set of frames by incorporating all additional frames that directly link to the first. From this expanded set of frames, we collect all example sentences. These are then combined to yield our set of literal input sentences I. We then build a metaphoric vocabulary S for the mapped source domain. This is done by extracting the lexical items provided by the MetaNet source domain, as well as from the lexical items from the FrameNet frame that is linked to the MetaNet source domain. To build the target vocabulary T, this process is repeated for words from the target domain of the MetaNet mapping.

Finally, we take the input sentences from the target domain (extracted from FrameNet examples), and replace the words from the target domain set T with masks. We then fill the masks with the best fit word from the source vocabulary using FitterBERT, a filter layer built on top of BERT(Devlin et al., 2019). FitterBERT is inspired by FitBERT (Havens and Stal, 2019), but is three orders of magnitude faster.

	Stowe et al. (2021)	**MetaNet Silver**
# Sentences	248k	360k
Unique Mappings	8.5k	650
Unique Domains	1k	550
Avg. Sent. Length (words)	8.2	23.6
Source	Gutenberg Corpus	FrameNet corpus

Table 1: Summary of the two datasets used for metaphor generation training.

Consider the example in Figure 1. On the left, we use the abstract NATIONS frame from MetaNet, map it to the respective POLITICAL LOCATION frame in FrameNet, then extract literal example sentences I and target vocabulary T. On the right, we collect candidate vocabulary S from the source domain in MetaNet and frame in FrameNet, and replace the target word from the literal input with appropriate source-domain vocabulary to yield a matching metaphoric paraphrase.

Note that this procedure functions as a reverse of the procedure from Section 3.1. They generate literal counterparts given the metaphor as the starting point, following the assumption that the language model will fill a more literal word into the context. We start with the literal input, and generate a metaphoric counterpart by constraining the vocabulary via known mappings and resources. This allows us to leverage the knowledge present in MetaNet and FrameNet to build metaphoric pairs motivated by conceptual metaphor theory. We refer to this dataset as the **MNS** (MetaNet silver) corpus.

Both datasets thus consist of semi-supervised paired sentences, annotated with the conceptual domains they evoke (one from FrameNet, one from MetaNet frames: these differ, but only slightly). A summary of both datasets is shown in Table 1.

Note that the MNS corpus has two key of advantages: first, the mappings are constrained to a set of hand-crafted metaphors from MetaNet. Second, the average sentence length is much longer, as the FrameNet annotations used cover a much more diverse set of sentences than the Gutenberg poetry corpus.

4 Methods

4.1 Seq2seq Models

For our free and controlled generation models, we use the T5 system for sequence to sequence generation (Raffel et al., 2020). This allows to implement control codes for metaphoricity and intended source/target mappings directly into the input texts. For an overview of the model inputs, see Table 2.

This process of including additional meta-information about the intended generation is similar to the language/task embedding paradigm, in which new languages and tasks can be handled by language models through directly appending relevant textual information(Ammar et al., 2016; Duong et al., 2017).

4.2 SOW-REAP Paraphrase Generation

We evaluate the SOW-REAP paraphrase model built to generate a diverse set of paraphrases for a given input (Goyal and Durrett, 2020). This model is naive with regard to metaphoricity: we anticipate the generated outputs should be strong with regard to fluency and semantic similarity, but not improve over the literal inputs with regard to metaphoricity.

The SOW-REAP model works by building a set of word re-orderings for the given input, then generating a set of diverse paraphrases for each re-ordering. To employ it in direct comparison, we select the single paraphrase over all re-orderings for a given input that had the highest intrinsic score, defined as the log probability of the output under a transformer-based model.

4.3 Free Generation

Free generation follows the use case of a user generating a metaphor from a given input, but without any knowledge of what would be an interesting, valid, or useful output. Free metaphor generation can be used to identify heretofore unknown connections between domains or fuel creative writing.

For free generation, we modify the context by adding a control code, "Activate metaphors". As we are fine-tuning a large pretrained T5 model, it may be the case that models can automatically generate metaphors from valid conceptual mappings as it has seen these metaphors before. However, without control, the model may also generate from incompatible domains, or otherwise incoherent metaphors, as metaphors tend to require some congruity between the concepts involved.

4.4 Controlled Generation

Controlled generation is intended to build paraphrases with explicit encoding of the semantics of intended metaphor. For our purposes, this is done by incorporating mappings based in CMT; incorporation of any type of metaphoric representations is theoretically possible. Controlled generation follows the use case where a user knows the relation they intend to capture, and needs the model to be consistent. This is necessary for longer text/story generation to have consistent metaphors, and in general to generate expressions for which finer-grained control of the semantics is required.

To incorporate controlled generation into T5, we include target and source information into the prefix, which then matches the format "Activate metaphors from TARGET to SOURCE:". Note that the relevant focus words are not marked: the model is free to evoke the metaphor flexibly, although as the training pairs vary only in the focus verb, in practice this is where changes are typically observed. As the model is always asked to activate metaphors, it learns a generalization over metaphoric expressions, but also additional signal regarding the input and output domains. Controlled generation constrains the model to specific domains, which increases the sparsity of training data for particular domains. However, using a large pre-trained language model as the base allows us to fine-tune on sparse data and still generate valid metaphoric expressions.

5 Human Evaluation

We develop both free and controlled models for the two datasets outlined in Section 3. We evaluate them on a hand-crafted test set which includes samples both within the data set they were trained on and from the unseen dataset.

5.1 Test Data

To evaluate model performance, we create a test set comprised of 250 literal/metaphoric pairs. We start with the 150 pairs from Stowe et al. (2021), which are taken from the Mohammad et al. (2016) corpus and the Gutenberg poetry corpus (Jacobs, 2018).

In addition, we add 100 paired sentences from the data collected in Section 3.2, which are first removed from the training dataThese are hand annotated for quality, ensuring that they contain valid metaphoric/literal pairs and source/target domain mappings. 50 were selected from the "narrow"

Context	Betty ushered the guests into the cottage.
Reference	Betty steered the guests into the cottage.
Domains	ushered:LEADERSHIP, steered:VEHICULAR_MOTION
Free Generation Context	Activate metaphors: Betty ushered the guests into the cottage.
Controlled Generation Context	Activate metaphors from LEADERSHIP IS VEHICULAR_MOTION: Betty ushered the guests into the cottage.

Table 2: Data inputs for T5 seq2seq metaphoric paraphrase generation.

Metric	SOW-REAP	Free			Ctrl			Gold
		MNS	Stowe	All	MNS	Stowe	All	
% not paraphrased	2.4	39.6	22.0	18.4	38.4	13.2	6.4	-
Fluency	3.176	3.461‡	3.474	**3.511**†	3.320	3.418	3.355	3.396
Sentence Similarity	3.246	3.648‡	3.676‡	**3.680**‡	3.458	3.573	3.460	3.474
Metaphoricity	2.278	2.348	2.490	2.424	2.430	**2.666**‡	2.593†	2.690

Table 3: Human evaluation scores (1-4) for each generation method. † ($.05 > p > .01$), ‡ ($p < .01$) over opposing model (Free/Ctrl)

metaphors and 50 were selected from the "broad" metaphors as annotated by MetaNet.

We use crowdsourcing to annotate each generated sentence for all models as well as the gold outputs. We first identify samples in which the output exactly matches the input; these samples are then excluded from further analysis. Following previous work in metaphor evaluation, we rate the generated outputs on three characteristics: (1) **fluency**, to evaluate the general grammatical quality of the output, (2) **semantic similarity**, to assess whether the output is a valid paraphrase, and (3) **metaphoricity**. For a full description of our crowdsourcing process, see Appendix A.

5.2 Analysis

Mean scores of all annotations for each model are shown in Table 3.[3] We included our gold standard test data in the human evaluation, as these metaphors are often difficult to understand, not perfect paraphrases, or contain other quirks of creative language. We find that the gold paraphrases have good fluency and semantic similarity scores, but aren't viewed as tremendously metaphoric. This is likely due to the relatively conventional nature of many of the gold metaphors: they are metaphoric in our linguistic analysis, as the metaphor evokes a connection between source and target domains, but they are frequent and normal in every day language, so crowdworkers were less likely to consider them good, novel metaphors.

For each model, we assess whether it produces significant improvements over the opposing version. Specifically, we are interested in whether the free models improve over the controlled and vice-versa. We evaluate using a paired t-test, setting our significance value at $p < .05$, while also noting where significance values were stronger ($p < .01$).

The SOW-REAP neural paraphrasing model performs relatively poorly. Somewhat surprisingly, it scores lowest on both fluency and semantic similarity, for which it should be strong. Its weak metaphoricity scores are to be expected: the model is designed to generate diverse paraphrases, but there is no reason for them to be metaphoric.

The models based on MetaNet data have better fluency and sentence similarity scores than SOW-REAP. However, metaphoricity is still low. The models trained on the Stowe et al. (2021) data have a stronger metaphoric signal. Combining both datasets yields small improvements in some cases, but the individual models outperform the combined in others. The key advantage of combining models is the coverage: combining models drastically reduces the percentage of non-paraphrased sentences.

Adding control to the metaphor generation models greatly improves performance with regard to metaphoricity, improving metaphoricity scores in all experiments, with significant gains of .176 and .171 for the Stowe and All training set models. However, the free models perform moderately better with regard to semantic similarity and fluency. Two of three fluency scores and all sentence similarity scores from Free models significantly outperformed the controlled versions. During training for free models, the models aren't constrained to learn from specific mapping-based vocabulary, and thus have greater freedom in generation, which can then

[3]Note that the evaluations are done after removing non-paraphrased instances, in which the hypothesis matched the context exactly. We evaluate other possible options in Appendix B.

be used to better match the original input. They generate words and phrases that better fit the context at the expense of metaphoricity. In some cases, they may even choose less metaphoric words which seem more natural, yielding better fluency and sentence similarity scores, but lowering metaphoricity. Controlled systems, on the other hand, are capable of using our understanding of conceptual metaphor theory to produce stronger metaphoric expressions. Their scores for fluency and sentence similarity are lower than the free models, but are still on par with the original gold references.

6 Automatic Evaluation

Evaluation of metaphor generation is a difficult task, as traditional metrics for machine translation and other tasks aim to enforce lexical and semantic similarity between the input and output sequences. As these metrics often rely on word overlap, valid metaphoric paraphrases may be punished for being overly creative. We thus implement a variety of standard evaluation metrics and evaluate their correlation with our gold standard annotations.

BLEU, METEOR, ROUGE Word and phrase overlap metrics from machine translation such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), and ROUGE (Lin, 2004) are often used to evaluate paraphrasing, despite noted weaknesses (Reiter and Belz, 2009; Reiter, 2018). We include them here to highlight their performance on creative language, and to compare them against human evaluations.

Translation Error Rate TER is another commonly used metric in machine translation, measuring the amount of correction that is necessary for a generated output to be valid (Snover et al., 2006).

SentBERT SentBERT (Reimers and Gurevych, 2019) provides sentence transformers to generate sentential vectors. These can be then compared using cosine distance to find semantic similarity. SentBERT has proven effective for a wide variety of similarity tasks, and should also be effective at determining paraphrase quality between literal and metaphoric sentences.

MoverScore MoverScore (Zhao et al., 2019) is a metric that uses BERT and Earth Mover Distance to measure similarity between two sentences. It uses contextual embeddings similar to SentBERT, and has the potential to better represent sense-specific meanings like those associated with metaphoricity.

Perplexity Transformer-based language models such as the GPT family (Radford et al., 2019; Brown et al., 2020) are extremely effective at producing text. The perplexity of a given sentence under the language model can be a proxy for sentence fluency. This approach promotes common words over rare, perhaps penalizing creativity, but can used as an evaluation metrics for generation systems (Chen et al., 2020; Bao et al., 2019).

Abstractness The notions of abstractness and concreteness have long been staples in metaphor detection systems (Dunn, 2013; Turney et al., 2011). We here use the abstract/concreteness ratings from Köper and Schulte im Walde (2017), which are based on Word2Vec embeddings (Mikolov et al., 2013). We evaluate the mean abstractness score, as well as the standard deviation, under the assumption that one of the main aspects of metaphoricity is the difference between abstractness levels of different conceptual domains within a sentence.

Novelty Classification We train a BERT regression model on the metaphoric novelty scores from Do Dinh et al. (2018). We score the generated sentence as the mean metaphoricity score over all words in the generated sentence.

Binary Metaphor Classification For binary metaphor classification, we use the DeepMet system of Su et al. (2020), which achieved best performance on the metaphor shared task (Leong et al., 2018). This model uses linguistics and contextual features in a siamese architecture, taking advantage of local and distant context. DeepMet functions at the word level; we score the generated sentence by taking the average number of words in the sentence classified as metaphoric.

We consider the generated output the **hypothesis**, the gold target metaphor as the **reference**, and the original literal input as the **context**. We compare each of these metrics to the crowdsourced annotations for all of our system outputs ($n = 1458$ unique hypotheses). For the similarity metrics, we evaluate using the hypothesis against the reference as well as against the context. For the others, they can intrinsically evaluate the hypothesis alone.

Pearson correlations are shown in Figure 2. Note that each metric has a specific goal, and thus we don't expect them to all perform well over all metrics. The word overlap metrics as well as SentBERT and MoverScore are designed to capture sentence similarity, perplexity is designed to capture sentence fluency, and the metaphoric models are designed to capture metaphoricity.

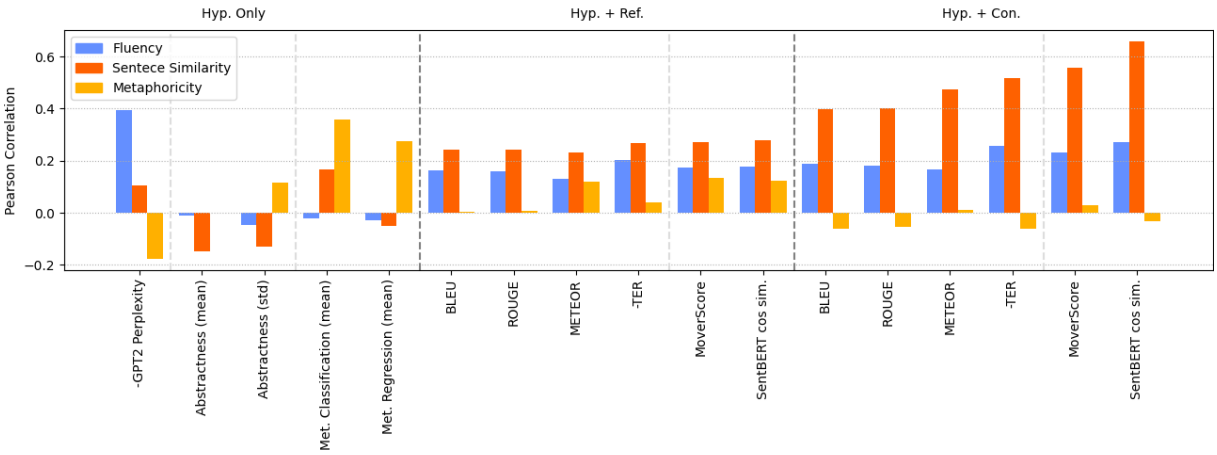

Figure 2: Pearson correlations between automatic evaluation metrics and crowdsourced labels.

No metric adequately captures all three aspects, although individual metrics excel at the components they are designed for. Peak correlations for fluency and metaphoricity are below .4, indicating these tasks are difficult to evaluate automatically in a way that is consistent with human evaluation. We further explore each component and the relevant metrics' performance:

6.1 Fluency

Perplexity under GPT-2 performs best for fluency, far outperforming any other metric, which is expected as this was the intended use of this metric. Semantic similarity metrics correlate with fluency, as a fluent output sentence is more likely to be similar to the original contexts and the reference sentences, both of which are fluent.

Note that while perplexity yields the best results for fluency, it correlates negatively with metaphoricity. This means that accurate evaluation of fluency will punish good metaphoric sentences. Perplexity thus cannot be used exclusively to evaluate generated metaphors, but rather requires corresponding metrics to evaluate the metaphoricity of the output.

6.2 Sentence Similarity

Both standard word overlap metrics and contextual embedding-based metrics correlate strongly with sentence similarity. We see much stronger correlations when comparing the generated hypothesis with the original input context. Our goal is to identify a sentence that semantically matches the context rather than the reference. Many possible metaphoric outputs could be suitable: comparing the hypothesis to a singular reference is less valuable, as it only reflects one possible option. Hypotheses can vary substantially from the reference and still be valid paraphrases of the context.

MoverScore and SentBERT compared to the context yield the best correlation. Specifically, SentBERT cosine similarity achieves a strong correlation value of .65, making it a valuable metric for evaluating this component. While these metrics all correlate to some degree with fluency, as fluent outputs are likely more similar to the original fluent context and references, they show no strong correlation with metaphoricity.

6.3 Metaphoricity

The mean abstractness of the generated outputs does not correlate significantly with metaphoricity. However, the standard deviation of the abstractness values performs better, supporting the idea that metaphoricity hinges upon variations in concrete and abstract terms. The binary classification model based on DeepMet has the strongest correlation with human metaphoricity scores. This is somewhat unexpected: the binary classification model averaged over the sentence correlates better than a regression model trained on metaphor novelty.

In summary, these metrics all capture different aspects of metaphoric paraphrase generation. Using one alone is insufficient, and can be misleading, particularly with regard to fluency and metaphoricity. We require a combination of perplexity under a language model, state-of-the-art metaphor classification, and embedding based semantic similarity metrics to capture the critical aspects of metaphoric paraphrase generation.[4]

[4]Generated outputs along with their human evaluation scores to are provided with the repository.

7 Conclusions and Future Work

We show that adding control to the metaphor generation process by means of conceptual domains improves the metaphoricity of generated paraphrases, but can decrease fluency and semantic similarity. Free generation systems are entirely viable, and may present advantages depending on the required task.

Second, we explore automatic evaluation metrics, showing that while previously employed metrics are capable of capturing some aspects of generation, there may be conflict between components, making multiple metrics necessary in order to best reflect the overall quality of generated outputs.

These results provide multiple ways forward for metaphor generation. With regard to free generation, we see that models are strong with regard to fluency and sentence similarity: future work thus requires a stronger signal for metaphoricity. This could be achieved by using improved datasets, or metaphor-specific generation models. For controlled generation, we explore only using domains from lexical resources as proxies for conceptual domains in CMT. Other theories of metaphor (including conceptual blends (Fauconnier and Turner, 1996), class-inclusion theory (Glucksberg, 2001), and structure-mapping (Gentner, 1983)) have yet to be explored for metaphor generation. With regard to evaluation, we've shown the strengths and weaknesses of some metrics, but none perform exceptionally well at this task overall. New metrics are needed; a metric that can harmonize syntactic fluency and the diversity of metaphoric expressions is necessary for rapid model development and evaluation.

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.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. Many Languages, One Parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, USA. Association for Computational Linguistics.

Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xin-yu Dai, and Jiajun Chen. 2019. Generating sentences from disentangled syntactic and semantic spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6008–6019, Florence, Italy. Association for Computational Linguistics.

Yuri Bizzoni and Shalom Lappin. 2018. Predicting human metaphor paraphrase judgments with deep neural networks. In *Proceedings of the Workshop on Figurative Language Processing*, pages 45–55, New Orleans, Louisiana, USA. Association for Computational Linguistics.

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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Asso-*

ciation for Computational Linguistics, pages 7929–7942, Online. Association for Computational Linguistics.

Alice Deignan. 2005. *Metaphor and Corpus Linguistics*. John Benjamins.

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, USA. 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, USA. Association for Computational Linguistics.

Jonathan Dunn. 2013. What metaphor identification systems can tell us about metaphor-in-language. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 1–10, Atlanta, Georgia. Association for Computational Linguistics.

Long Duong, Hadi Afshar, Dominique Estival, Glen Pink, Philip Cohen, and Mark Johnson. 2017. Multilingual semantic parsing and code-switching. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 379–389, Vancouver, Canada. Association for Computational Linguistics.

Elozino Egonmwan and Yllias Chali. 2019. Transformer and seq2seq model for paraphrase generation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 249–255, Hong Kong, China. Association for Computational Linguistics.

Gilles Fauconnier and Mark Turner. 1996. Blending as a central process of grammar. In Adele Goldberg, editor, *Conceptual Structure, Discourse, and Language*. Cambridge University Press.

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

Sam Glucksberg. 2001. *Understanding Figurative Language*. Oxford University Press, London, England, UK.

Tanya Goyal and Greg Durrett. 2020. Neural syntactic preordering for controlled paraphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 238–252, Online. Association for Computational Linguistics.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

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.

Maximilian Köper and Sabine Schulte im Walde. 2017. Improving verb metaphor detection by propagating abstractness to words, phrases and individual senses. In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 24–30, Valencia, Spain. Association for Computational Linguistics.

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

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

Chee Wee (Ben) Leong, Beata Beigman Klebanov, Chris Hamill, Egon Stemle, Rutuja Ubale, and Xianyang Chen. 2020. A report on the 2020 VUA and TOEFL metaphor detection shared task. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 18–29, Online. Association for Computational Linguistics.

Chee Wee (Ben) Leong, Beata Beigman Klebanov, and Ekaterina Shutova. 2018. A report on the 2018 VUA metaphor detection shared task. In *Proceedings of the Workshop on Figurative Language Processing*, pages 56–66, New Orleans, Louisiana, USA. Association for Computational Linguistics.

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.

Zichao Li, Xin Jiang, Lifeng Shang, and Qun Liu. 2019. Decomposable neural paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3403–3414, Florence, Italy. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Kathleen R. McKeown. 1983. Paraphrasing questions using given and new information. *American Journal of Computational Linguistics*, 9(1):1–10.

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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Aaditya Prakash, Sadid A. Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2016. Neural paraphrase generation with stacked residual LSTM networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2923–2934, Osaka, Japan. The COLING 2016 Organizing Committee.

Lihua Qian, Lin Qiu, Weinan Zhang, Xin Jiang, and Yong Yu. 2019. Exploring diverse expressions for paraphrase 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 3173–3182, Hong Kong, China. Association for Computational Linguistics.

Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of Empirical Methods in Natural Langyuage Process (EMNLP)*, pages 142–149. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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 Machine Learning Research*, 21(140):1–67.

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.

Ehud Reiter. 2018. A structured review of the validity of bleu. *Computational Linguistics*, 44(3):393–401.

Ehud Reiter and Anja Belz. 2009. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *In Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Kevin Stowe, Tuhin Chakrabarty, Violent Peng, Smaranda Muresan, and Iryna Gurevych. 2021. Generating metaphors with conceptual mappings. *ArXiv*, 2005.14165.

Kevin Stowe, Leonardo Ribeiro, and Iryna Gurevych. 2020. Metaphoric paraphrase generation. *ArXiv*, 2005.14165.

Chuandong Su, Fumiyo Fukumoto, Xiaoxi Huang, Jiyi Li, Rongbo Wang, and Zhiqun Chen. 2020. DeepMet: A reading comprehension paradigm for token-level metaphor detection. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 30–39, Online. Association for Computational Linguistics.

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

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.

Peter Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 680–690, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Su Wang, Rahul Gupta, Nancy Chang, and Jason Baldridge. 2019. A task in a suit and a tie: Paraphrase generation with semantic augmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7176–7183.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2010. Paraphrase generation as monolingual translation: Data and evaluation. In *Proceedings of the 6th International Natural Language Generation Conference*.

Qiongkai Xu, Juyan Zhang, Lizhen Qu, Lexing Xie, and Richard Nock. 2018. D-page: Diverse paraphrase generation. *ArXiv*, 1808.04364.

Qian Yang, Zhouyuan Huo, Dinghan Shen, Yong Cheng, Wenlin Wang, Guoyin Wang, and Lawrence Carin. 2019. An end-to-end generative architecture for paraphrase 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 3132–3142, 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, USA. Association for Computational Linguistics.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. 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 563–578, Hong Kong, China. Association for Computational Linguistics.

A Crowdsourcing

For crowdsourcing, we employed Amazon Mechanical Turk. We defined our three tasks with brief definitions, and ran a set of 10 instances for which we had gold scores for up to 100 crowdworkers to complete. We initially filtered the workers by those with at least a 90% approval rate, whose origin was an English speaking country[5]. We then evaluated those workers for competence. We selected only workers that completed at least 8 instances, with an average error of < 1 per instance from the gold score. We also excluded all users were had answers with an error greater than one, to exclude those who had poor understanding of the task. This yielded a set of users which we then white-listed to complete the final evaluation. This ensures high quality annotations with the need for including test instances in the final set.

The definitions for each metric are provided are below:

- Fluency: Select how grammatical/fluent the sentence is from 1 (completely incomprehensible) to 4 (fluent, grammatical English). Don't worry about capitalization or punctuation. Just try to determine if you heard the sentence, how grammatical would you consider it? Some word choices may be strange, but try to focus on the syntax. If the sentence is grammatically valid, mark 4. If it has some minor errors, mark 3. If it has multiple errors, or is very hard to understand, mark 2. If it is completely unintelligible, mark 1.

- Sentence Similarity: Select how similar the meanings of the two sentences are from 1 (completely unrelated) to 4 (good paraphrase). Ignore punctuation/extra spacing/capitalization. We care only about the meaning/semantics. Do the sentences have the same meaning? If so, select 4. Some sentences may be paired with abstract or metaphoric paraphrases: as long as they have the same meaning, mark as 4. If the sentences are similar but mean something different, mark 3. If the sentences are not very similar but have some similar ideas, mark as 2. If the sentences are completely unrelated, or incomprehensible, mark as 1.

[5] US, CA, GB, NZ, AU, IE

Metric	α
Fluency	.423
Sentence Similarity	.364
Metaphoricity	.338

Table 4: Interannotator Agreement (Krippendorff's α) for each metric.

- Metaphoricity: Select how metaphoric each sentence, from 1 being the most literal/least metaphoric to 4 being the strongest, most novel metaphors. Metaphors involve using language from one domain to describe another. Good metaphors using novel language to connect two concepts, often in creative and interesting ways. Literal sentences describe the world as it is, and are typically more basic and concrete. You can ignore grammar/punctuation: try to just assess the meaning of the sentence.

We paid workers .08$ USD per task, aiming for approximately 10$ per hour. Each task was completed by five annotators. Inter-annotator agreements rates are shown in Table 4. Despite reflecting relatively low agreement, these are comparable with previous agreement scores for crowdsourced metaphoricity evaluation, which lie between .16-.49 α for crowdsourcing, and approximately .5 for expert annotation (Do Dinh et al., 2018; Chakrabarty et al., 2021; Stowe et al., 2021). These difficulties are likely due to the unique syntactic and semantic nature of metaphoricity; more work on better human evaluation is necessary.

B Evaluating Matching Hypothesis

We initially removed all outputs in which the hypothesis exactly matched the context, as these indicate a failure to generate a paraphrase. We experiment with two alternative approaches. First, we remove all sentences from evaluation for which any model matches the input, thus allowing for equal comparison across all models for a smaller dataset (Table 5). Second, we automatically score all sentences that exactly matched the context with their theoretically expected scores: maximal fluency and sentence similarity, with minimal metaphoricity (Table 6).

These results match our expectations, and confirm the patterns reported in Section 5.2. The controlled models generate better metaphors, while the free models perform better with regard to fluency and sentence similarity. Giving default scores to hypotheses that match the context yields strong fluency and sentence similarity scores scores, while metaphoricity suffers for models that can't generate novel paraphrases. When the model has strong coverage (ie. the fully controlled model), metaphoricity scores remain strong.

Metric	SOW-REAP	Free			Ctrl			Gold
		MNS	Stowe	All	MNS	Stowe	All	
% not paraphrased	1.0	33.2	30.6	31.9	22.5	17.1	19.8	-
Fluency	2.998	3.432	3.440	**3.507**	3.359	3.394	3.313	3.243
Semantic Similarity	3.206	**3.747**	2.732	2.729	3.559	3.682	3.549	3.449
Metaphoricity	2.366	2.387	2.427	2.374	2.508	2.485	**2.543**	2.269

Table 5: Human evaluation scores (1-4) for each generation method, using only samples for which all models generated a hypothesis that differed from the context ($n = 109$).

Metric	SOW-REAP	Free			Ctrl			Gold
		MNS	Stowe	All	MNS	Stowe	All	
% not paraphrased	1.0	33.2	30.6	31.9	22.5	17.1	19.8	-
Fluency	3.196	**3.682**	3.591	3.648	3.583	3.496	3.418	3.396
Sentence Similarity	3.264	**3.792**	3.748	3.770	3.668	3.630	3.513	3.474
Metaphoricity	2.247	1.794	2.158	2.024	1.877	**2.444**	2.439	2.690

Table 6: Human evaluation scores (1-4) for each generation method, giving default scores to hypotheses that matched the context.