MOVER: Mask, Over-generate and Rank for Hyperbole Generation

Anonymous ACL submission

Abstract

Despite being a common figure of speech, hyperbole is under-researched in Figurative Language Processing. In this paper, we tackle the challenging task of hyperbole generation to transfer a literal sentence into its hyperbolic paraphrase. To address the lack of available hyperbolic sentences, we construct HYPO-XL, the first large-scale hyperbole corpus containing 17,862 hyperbolic sentences in a non-trivial way. Based on our corpus, we propose an unsupervised method for hyperbole generation that does not require parallel literal-hyperbole pairs. During training, we fine-tune BART (Lewis et al., 2020) to infill masked hyperbolic spans of sentences from HYPO-XL. During inference, we mask part of an input literal sentence and over-generate multiple possible hyperbolic versions. Then a BERT-based ranker selects the best candidate by hyperbolicity and paraphrase quality. Automatic and human evaluation results show that our model is effective at generating hyperbolic paraphrase sentences and outperforms several baseline systems.

1 Introduction

Hyperbole is a figure of speech that deliberately exaggerates a claim or statement to show emphasis or express emotions. If a referent has a feature X, a hyperbole exceeds the credible limits of fact in the given context and presents it as having more of that X than warranted by reality (Claridge, 2010). Take the following example, "*I won't wait for you: it took you centuries to get dressed.*" It over-blows the time for someone to get dressed with a single word "*centuries*" and thus creates a heightened effect. From a syntactic point of view, Claridge (2010) classifies hyperbole into word-level, phrase-level and clause-level types, and conclude that the former two types are more common in English. Although hyperbole is considered as the second most frequent figurative device (Kreuz and Roberts, 1993), it has received less empirical attention in the NLP community. Recently Tian et al. (2021) addressed the generation of *clause-level* hyperbole. In this paper, we instead focus on *word-level* and *phrase-level* hyperbole, which can be unified as span-level hyperbole.

To tackle the hyperbole generation problem we need to address three main challenges: 1) the lack of training data that either consists of large-scale hyperbolic sentences or literal-hyperbole pairs, which are necessary to train an unsupervised or supervised model; 2) the tendency of generative language models to produce literal text rather than hyperbolic one; 3) trade-off between content preservation and hyperbolic effect of the generated sentences.

In order to address the above challenges, we propose **MOVER** (**M**ask, **OVE**r-generate and **R**ank), an unsupervised approach to generating hyperbolic paraphrase from literal input. Our approach does not require parallel data for training, thus alleviating the issue of scarce data. Still, we need a non-parallel corpus containing as much hyperbolic sentences as possible. To this end, we first build a large-scale hyperbole corpus HYPO-XL in a weakly supervised way.

Based on the intuition that the hyperbolic effect of a sentence is realized by a single word or phrase within it, we introduce a sub-task of hyperbolic span extraction. We identify several possible n-grams of a hyperbolic sentence that can cause the hyperbolic bent with syntactic and semantic features. We apply this masking approach to sentences in HYPO-XL and teach a pretrained seq2seq transformer, BART (Lewis et al., 2020), to infill the words in missing hyperbolic spans. This increases the probability of generating hyperbolic texts instead of literal ones. During inference, given a single literal sentence, our system provides multiple masked versions for inputs to BART and generates potential hyperbolic sentences accordingly. To select the best one for output, we leverage a BERT-based ranker to achieve a satisfying trade-off between hyperbolicity and paraphrase quality.

Our contributions are three-fold: 1) We construct the first large-scale hyperbole corpus HYPO-XL in a non-trivial way. The corpus will be released[1] and contribute to the Figurative Language Processing (FLP) community by facilitating the development of computational study of hyperbole. 2) We propose an unsupervised approach for hyperbole generation that falls into the "overgenerate-and-rank" paradigm (Heilman and Smith, 2009). 3) We benchmark our system against several baselines and we compare their performances by pair-wise manual evaluations to demonstrate the effectiveness of our approach.

2 HYPO-XL: Hyperbole Corpus Collection

The availability of large-scale corpora can facilitate the development of figurative language generation with pretrained models, as is shown by Chakrabarty et al. (2020c) on simile generation and Chakrabarty et al. (2021) on metaphor generation. However, datasets for hyperbole are scarce. Troiano et al. (2018) built an English corpus HYPO containing 709 triplets $[hypo, para, non\_hypo]$, where $hypo$ refers to a hyperbolic sentence, $para$ denotes the literal paraphrase of $hypo$ and non_hypo means a non-hyperbolic sentence that contains the same hyperbolic word or phrase as $hypo$ but with a literal connotation. The size of this dataset is too small to train a deep learning model for hyperbole detection and generation. To tackle the lack of hyperbole data, we propose to enlarge the hyperbolic sentences of HYPO in a weakly supervised way and build a large-scale corpus of 17,862 hyperbolic sentences, namely HYPO-XL. We would like to point out that this is a *non-parallel* corpus containing only hyperbolic sentences without their paraphrase counterparts, because our hyperbole generation approach (Section 3) does not require parallel training data.

The creation of HYPO-XL consists of two steps: 1) We first train a BERT-based binary classifier on HYPO and retrieve possible hyperbolic sentences from an online corpus. 2) We manually label a subset of the retrieved sentences, denoted HYPO-L, and retrain our hyperbole detection model to identify hyperbolic sentences from the same retrieval corpus with higher confidence.

[1]The data has been uploaded as supplementary material of this submission.

2.1 Automatic Hyperbole Detection

Hyperbole detection is a supervised binary classification problem where we predict whether a sentence is hyperbolic or not (Kong et al., 2020). We fine-tune a BERT-base model (Devlin et al., 2019) on the hyperbole detection dataset HYPO (Troiano et al., 2018). In experiment, we randomly split the data into 567 (80%) hyperbolic sentences, with their literal counterparts ($para$ and non_hypo) as negative samples, in training set and 71 (10%) in development set and 71 (10%) in test set. Our model achieves an accuracy of 80% on the test set, which is much better than the highest reported accuracy (72%) of traditional algorithms in Troiano et al. (2018).

Once we obtain this BERT-based hyperbole detection model, the next step is to retrieve hyperbolic sentences from a corpus. Following Chakrabarty et al. (2020a), we use Sentencedict.com,[2] an online sentence dictionary as the retrieval corpus. We remove duplicate and incomplete sentences (without initial capital) in the corpus, resulting in a collection of 767,531 sentences. Then we identify 93,297 (12.2 %) sentences predicted positive by our model as pseudo-hyperbolic.

2.2 HYPO-L: Human Annotation of Pseudo-labeled Data

Due to the small size of training set, pseudo-labeled data tend to have lower confidence score (i.e., the prediction probability). To improve the precision of our model,[3] we further fine-tune it with our human-annotated data, namely HYPO-L. We randomly sample 5,000 examples from the 93,297 positive predictions and invite students with proficiency in English to label them as hyperbolic or not. For each sentence, two annotators provide their judgements. We only keep items with unanimous judgments (i.e. both of the two annotators mark the sentence as hyperbolic or non-hyperbolic) to ensure the reliability of annotated data. In this way, 3,226 (64.5%) out of 5,000 annotations are left in HYPO-L. This percentage of unanimous judgments (i.e., raw agreement, RA) is comparable to 58.5% in the creation of HYPO (Troiano et al., 2018). To be specific, HYPO-L consists of 1,007 (31.2%) hyperbolic sentences (positive samples) and 2,219 (68.8%) literal ones (negative samples).

[2]https://sentencedict.com/

[3]Given the massive hyperboles in the "wild" (i.e., the retrieval corpus) we do not pursue recalling more hyperboles at the risk of hurting precision (Zhang et al., 2021).

Measurement	Value
% Non-hypo	6%
# Avg hypo span tokens	2.23
% Long hypo spans (> 1 token)	37%
# Distinct hypo spans	85
# Distinct POS-ngrams of hypo spans	39

Table 1: Statistics of 100 random samples from HYPO-XL, of which 6 are actually non-hyperboles ("Non-hypo"). The statistics of hyperbolic text spans ("hypo span") are calculated for the rest 94 real hyperboles.

We continue to train the previous HYPO-fine-tuned BERT on HYPO-L and the test accuracy is 80%,[4] which we consider as an acceptable metric for hyperbole detection. Finally we apply the BERT-based detection model to the retrieval corpus again and retain sentences whose prediction probabilities for positive class exceed a certain threshold.[5] This results in HYPO-XL, a large-scale corpus of 17,862 (2.3%) hyperbolic sentences. We provide a brief comparison of HYPO, HYPO-L and HYPO-XL in Appendix A to clarify the data collection process.

2.3 Corpus Analysis

Since HYPO-XL is built in a weakly supervised way with only a few human labeled data samples, we conduct a quality analysis to investigate how many sentences in the corpus are *actually* hyperbolic. We randomly sample 100 instances from HYPO-XL and manually label them as hyperbole or non-hyperbole. Only six sentences are not hyperbole. This precision of 94% is on par with 92% on another figurative language corpus of simile (Zhang et al., 2021). Actually we can tolerate a bit noise in the corpus since the primary goal of HYPO-XL is to facilitate hyperbole *generation* instead of *detection*, and a small proportion of non-hyperbole sentences as input will not harm our proposed method.[6] The cost of manually filtering out non-hyperboles in the corpus would be too high for us. Table 1 shows the statistics of hyperbolic text spans (defined in Section 3.1) for the rest 94 real hyperboles. We also provide additional analyses in Appendix A.

[4] We separate 10% data of HYPO-L for development and another 10% for testing.

[5] Based on manual inspection of predicted results, we set the threshold as 0.8 to trade-off between precision and recall.

[6] We further explain the reason in Section 3.2

3 Hyperbole Generation

We propose an unsupervised approach to generate hyperbolic paraphrase from a literal sentence with BART (Lewis et al., 2020) such that we do not require parallel literal-hyperbole pairs. An overview of our hyperbole generation pipeline is shown in Figure 1. It consists of two steps during training: 1) **Mask** — Given a hyperbolic sentence from HYPO-XL, we identify multiple text spans that can possibly produce the hyperbolic meaning of a sentence, based on two features (POS n-gram and unexpectedness score). For each identified text span, we replace it with the $[\text{MASK}]$ token to remove hyperbolic attribute of the input. N text spans will result in N masked inputs, respectively. 2) **Infill** — We fine-tune BART to fill the masked spans of input sentences. The model learns to generate hyperbolic words or phrases that are pertinent to the context.

During inference, there are three steps: 1) **Mask** — Given a literal sentence, we apply POS-ngram-only masking to produce multiple input sentences. 2) **Over-generate** — BART generates one sentence from a masked input, resulting in multiple candidates. 3) **Rank** — Candidates are ranked by their hyperbolicity and relevance to the source literal sentence. The one with highest score is selected as the final output.

We dub our hyperbole generation system **MOVER** (**M**ask, **OVE**r-generate and **R**ank). We apply masking technique to map both the hyperbolic (training input) and literal (test input) sentences into a same "space" where the masked sentence can be transformed into hyperbole by BART. It falls into the "overgenerate-and-rank" paradigm (Heilman and Smith, 2009) since many candidates are available after the generation step. The remainder of this section details the three main modules: hyperbolic span masking (Section 3.1), BART-based span infilling (Section 3.2) and the hyperbole ranker (Section 3.3).

3.1 Mask: Hyperbolic Span Masking

We make a simple observation that the hyperbolic effect of a sentence is commonly localized to a single word or a phrase, which is also supported by a corpus-based linguistic study on hyperbole (Claridge, 2010). For example, the word *marathon* in *"My evening jog with Bill turned into a **marathon**"* overstates the jogging distance and causes the sentence to be hyperbolic. This inspires us to leverage the "delete-and-generate" strategy (Li et al.,

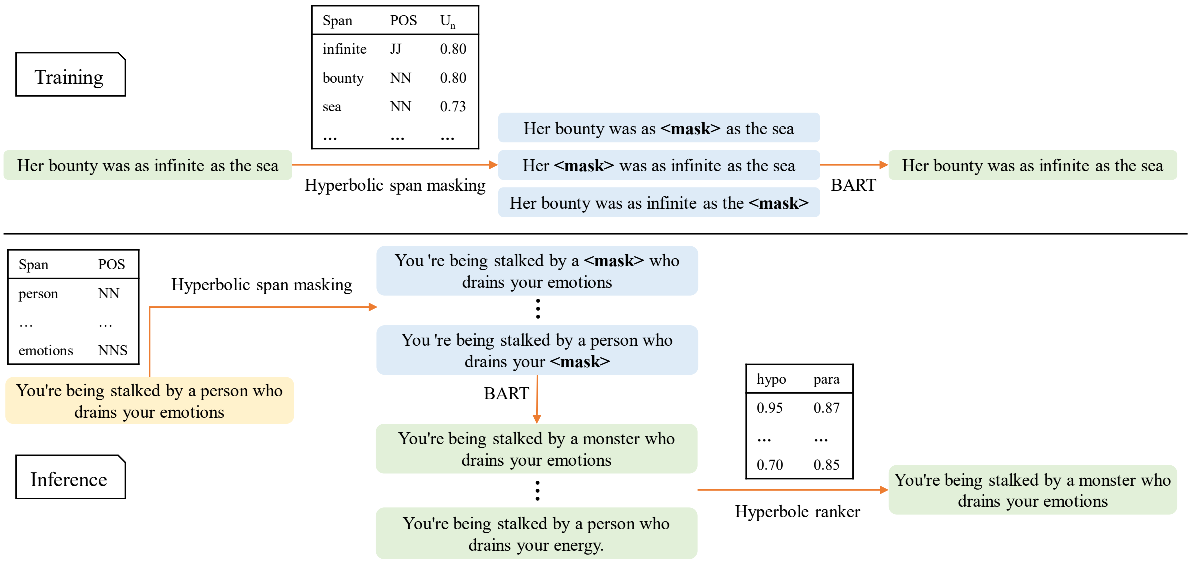

Figure 1: Overview of our approach to unsupervised hyperbole generation. Literal sentences are in yellow boxes, masked sentences are in blue boxes and hyperbolic sentences are in green boxes.

2018) for hyperbole generation. Concretely, a literal sentence can be transformed into its hyperbolic counterpart via hyperbolic span extraction and replacement. We propose to extract hyperbolic spans based on POS n-gram (syntactic) and unexpectedness (semantic) features.

POS N-gram We extract POS n-gram patterns of hyperbole from the training set of HYPO dataset[7] and obtain 262 distinct POS n-grams. As a motivating example, the following three hyperbolic spans, *"faster than light"*, *"sweeter than honey"*, *"whiter than snow"*, share the same POS n-gram of "JJR+IN+NN".

Unexpectedness Hyperbolic spans are less coherent with the literal contexts and thus their vector representations are distant from the context vectors. Troiano et al. (2018) have verified this intuition with the unexpectedness metric. They define the unexpectedness score U_s of a sentence s with the token sequence $\{x_0, x_1, ..., x_N\}$ as the average cosine distance among all of its word pairs.

$$U_s = \underset{i,j \in [0,N], i \neq j}{average} (cosine\_distance(v_i, v_j)) \quad (1)$$

where v_i denotes the word embedding vector of token x_i. Similarly, we define the unexpectedness score U_n of an n-gram $\{x_k, x_{k+1}, ..., x_{k+n-1}\}$ in a sentence s as the average cosine distance among word pairs that consist of one word inside the n-gram and the other outside.

$$U_n = \underset{\substack{i \in [k, k+n-1] \\ j \in [0, k-1] \cup [k+n, N]}}{average} (cosine\_distance(v_i, v_j)) \quad (2)$$

Text spans with higher unexpectedness scores tend to be hyperbolic. We provide an illustration of the unexpectedness score in Appendix B.

For the masking step during training, we extract all text spans in the original input hyperbolic sentences that match one of the hyperbolic POS n-grams. Then we rank them by their unexpectedness scores and choose top-3 items as the masked spans.[8] For the masking step during inference, we simply mask all the spans that match hyperbolic POS n-grams, since the span unexpectedness score is not applicable to a literal input. We evaluate the accuracy of our hyperbolic span masking approach on the development set of HYPO dataset. The proportion of exact match (EM) (Rajpurkar et al., 2016) between our top-3 masked spans with the human-labeled spans is 86%, which shows that our simple method based on the above-mentioned hand-crafted features is effective for the task of hyperbolic span extraction.

[7] The hyperbolic spans are not explicitly provided in the HYPO dataset, so we take the maximum word overlap between *hypo* and *non_hypo* (Section 2) as the hyperbolic spans.

[8] This means that at least 2/3 of the identified spans should not be hyperbolic, but this will not harm the training of our hyperbole generation model, which is explained in Section 3.2

3.2 Over-generate: Hyperbolic Text Infilling with BART

In order to generate hyperbolic and coherent text from the masked span, we leverage the text span infilling ability of BART (Lewis et al., 2020), a pretrained sequence2sequence model with a denoising autoencoder and an autoregressive autodecoder. During its pretraining, it learns to reconstruct the corrupted noised text. One of the noising transformations is random span masking, which teaches BART to predict the multiple tokens missing from a span. During our training process, we fine-tune BART by treating the masked hyperbolic sentence as the encoder source and the original one as the decoder target. This can change the probability distribution when decoding tokens and increase the chance of generating a hyperbolic, rather than literal, text span conditioned on the context. During inference, BART fills the masked span of a literal sentence with possible hyperbolic words.

Note that if the masked span of an input sentence is actually not hyperbolic, then fine-tuning on this example will just enhance the reconstruction ability of BART, which will not exert negative effects on hyperbole generation. This can give rise to our tolerance for non-hyperbolic sentences in the training corpus (Section 2.3) and non-hyperbolic masked span (Section 3.1).

3.3 Rank: Hyperbole Ranker

Recall that for each literal input during inference, we apply POS-ngram-based masking, produce different masked versions of the sentence, and generate multiple output candidates. Obviously, not all masking spans are suitable for infilling hyperbolic words due to the noise of masking. To select the best candidate for final output, we introduce a hyperbole ranker which sorts candidate sentences by their degree of hyperbolicity and relevance to the source inputs. For evaluation of hyperbolicity, we leverage the BERT-based hyperbole detection model fine-tuned on HYPO and HYPO-L (Section 2.2) to assign a hyperbole score (i.e., prediction probability) for every candidate. For the evaluation of content preservation, we train a pair-wise model to predict whether the hyperbolic sentence A is a paraphrase of a literal sentence B. To this end, we use the distilled RoBERTa-base model checkpoint[9] pretrained on large scale paraphrase data provided by Sentence-Transformer (Reimers and Gurevych, 2019). It calculates the cosine similarity between the literal input and the candidate as the paraphrase score. We fine-tune the checkpoint on the training set of HYPO dataset, where we treat the pairs of $hypo$ and $para$ as positive examples, and pairs of $hypo$ and non_hypo as negative examples (Section 2). The accuracy on test set is 93%.

Now that we obtain the hyperbole score $hypo(c)$ and the paraphrase score $para(c)$ for candidate c, we propose an intuitive scoring function $score(\cdot)$ as below:

$$score(s) = \begin{cases} hypo(s) & para(s) \in (\gamma, 1-\epsilon) \\ 0 & \text{else} \end{cases} \tag{3}$$

Here we filter out a candidate if its paraphrase score is lower than a specific threshold γ or it is almost the same as the original input (i.e., the paraphrase score is extremely close to 1). For diversity purposes, we do not allow our system to simply copy the literal input as its output. We then rank the remaining candidates according to their hyperbole score and select the best one as the final output.[10]

4 Experiments

There are no existing models applied to the task of word-level/phrase-level hyperbole generation. To compare the quality of the generated hyperboles, we benchmark our MOVER system against three baseline systems adapted from related tasks.

4.1 Baseline Systems

Retrieve (R1) Following Nikolov et al. (2020), we implement a simple information retrieval baseline, which retrieves the closest hyperbolic sentence as the output (i.e., the highest cosine similarity) from HYPO-XL, using the hyperbole paraphrase detection model $para(\cdot)$ in Section 3.3. The outputs of this baseline system should be hyperbolic yet have limited relevance to the input.

Retrieve, Replace and Rank (R3) We first retrieve the top-5 most similar sentences from HYPO-XL like the R1 baseline. Then we apply hyperbolic span extraction in Section 3.1 to find 3 text spans for each retrieved sentence. We replace the text

[9] https://huggingface.co/sentence-transformers/paraphrase-distilroberta-base-v1

[10] If all candidates are filtered out by their paraphrase scores (i.e. they all have the zero final scores), we will select the one with the highest hyperbole score among all candidates.

spans in a literal input sentence with retrieved hyperbolic spans if two spans share the same POS n-gram. Since this replacement method may result in multiple modified sentences, we select the best one with the hyperbole ranker in Section 3.3. If there are no matched text spans, we fall back to R1 baseline and return the most similar retrieved sentence verbatim. In fact, this baseline substitutes the BART generation model in MOVER system with a simpler retrieval approach, which can demonstrate the hyperbole generation ability of BART.

BART Inspired by Chakrabarty et al. (2020c), we replace the text infilling model in Section 3.2 with a non-fintuned off-the-shelf BART,[11] because BART has already been pretrained to predict tokens from a masked span.

4.2 Implementation Details

We use 16,075 (90%) samples in HYPO-XL for training our MOVER system and the rest 1,787 sentences for validation. For POS Tagging in Section 3.1 we use Stanford CoreNLP (Manning et al., 2014). For the word embedding we use 840B 300-dimension version of GloVe vectors (Pennington et al., 2014). For BART in Section 3.2 we use the BART-base checkpoint instead of BART-large due to limited computing resources and leverage the implementation by Huggingface (Wolf et al., 2020). We fine-tune pretrained BART for 16 epochs. For parameters of the hyperbole ranker in Section 3.3, we set $\gamma = 0.8$ and $\epsilon = 0.001$ by manual inspection of the ranking results on the development set of HYPO dataset.

4.3 Evaluation Criteria

Automatic Evaluation BLEU (Papineni et al., 2002) reflects the lexical overlap between the generated and the ground-truth text. BERTScore (Zhang et al., 2019a) computes the similarity using contextual embeddings. These are common metrics for text generation. We use the 71 literal sentences ($para$) in the test set of HYPO dataset as test inputs and their corresponding hyperbolic sentences ($hypo$) as gold references. We report the BLEU and BERTScore metrics for generated sentences compared against human written hyperboles.

[11] We also tried to fine-tune BART on the 567 literal-hyperbole pairs from the training set of HYPO dataset in an end-to-end supervised fashion, but the model just copy the input for all instances (same as COPY in Table 2) and is unable to generate meaningful output due to small amount of training data. Besides, we test the performance of a BART-based paraphrase generation model, which is BART finetuned on QQP (Wang et al., 2018) and PAWS (Zhang et al., 2019c) datasets. We still find that 50% of the outputs from the paraphrase model just copy the input. Therefore we do not consider these two BART-based systems hereafter.

System	BLEU	BERTScore
R1	2.02	0.229
R3	33.25	0.520
BART	33.57	0.596
MOVER	**39.43**	**0.624**
w/o para score	39.22	0.604
w/o hypo ranker	34.83	0.610
COPY	<u>51.69</u>	<u>0.711</u>

Table 2: Automatic evaluation results on the test set of HYPO dataset.

Human Evaluation Automated metrics are not reliable on their own for evaluating methods to generate figurative language (Novikova et al., 2017) so we also conduct pair-wise comparisons manually (Shao et al., 2019). We evaluate the generation results from the 71 testing literal sentences. Each pair of texts (ours vs. a baseline / human reference) is given preference (win, lose or tie) by five people with proficiency in English. We use a set of four criteria adapted from Chakrabarty et al. (2021) to evaluate the generated outputs: 1) **Fluency (Flu.)**: Which sentence is more fluent and grammatical? 2) **Hyperbolicity (Hypo.)**: Which sentence is more hyperbolic? 3) **Creativity (Crea.)**: Which sentence is more creative? 4) **Relevance (Rel.)**: Which sentence is more relevant to the input literal sentence?

4.4 Results

Automatic Evaluation Table 2 shows the automatic evaluation results of our system compared to different baselines. MOVER outperforms all three baselines on these two metrics. However, BLEU and BERTScore are far not comprehensive evaluation measures for our hyperbole generation task, since there are only a few modifications from literal to hyperbole and thus there is a lot of overlap between the generated sentence and the source sentence. Even a naive system (COPY in Table 2) that simply returns the literal input verbatim as output (Krishna et al., 2020) can achieve the highest performance. As a result, automatic metrics are not suitable for evaluating models that tend to copy input as output.

MOVER vs.	Flu.		Hypo.		Crea.		Rel.	
	W%	L%	W%	L%	W%	L%	W%	L%
R1	**79.7**	1.7	**52.4**	47.6	33.9	**66.1**	**94.2**	4.3
R3	**35.8**	11.3	**52.5**	36.1	**50.0**	38.5	**52.6**	29.8
BART	**26.2**	19.7	**67.7**	11.3	**61.0**	10.2	**49.2**	31.7
HUMAN	**22.0**	18.6	16.7	**81.8**	14.3	**84.3**	**46.8**	37.1

Table 3: Pairwise comparison between MOVER and other baseline systems. Win[W]% (Lose[L]%) is the percentage of MOVER considered better (worse) than a baseline system. The rest are ties.

System	Sentence	F.	H.	C.	R.
LITERAL	Being out of fashion is very bad.	-	-	-	-
MOVER	Being out of fashion is *sheer hell.*	-	-	-	-
R1	*Their music will never go* out of fashion.	T	**W**	L	**W**
R3	Being out of fashion is *richly* bad.	T	**W**	**W**	T
BART	Being out of fashion is very *difficult*.	T	**W**	**W**	T
HUMAN	*Better be out of the world than* out of the fashion.	**W**	**W**	L	**W**

Table 4: Pairwise evaluation results (Win[W], Lose[L], Tie[T]) in terms of **F**luency, **H**yperbolicity, **C**reativity and **R**elevance between MOVER and generated outputs of baseline systems. Changed text spans are in *italic*. More examples are in Appendix C.

Human Evaluation The inter-annotator agreement of raw human evaluation results in terms of Fleiss' kappa (Fleiss, 1971) is 0.212, which indicates fair agreement (Landis and Koch, 1977). We take a conservative approach and only consider items with an absolute majority label, i.e., at least three of the five labelers choose the same preference (win/lose/tie). There are 61 (86%) items on average left for each baseline-criteria pair that satisfy this requirement. On this subset of items, Fleiss' Kappa increases to 0.278 (fair agreement). This degree of agreement is acceptable compared to other sentence revision tasks (e.g., 0.322 by Tan and Lee (2014) and 0.263 by Afrin and Litman (2018)) since it is hard to discern the subtle changing effect caused by local revision.

The annotation results in Table 3 are the absolute majority vote (majority $>= 3$) from the 5 annotators for each item. Results show that our model mostly outperforms (Win% $>$ Lose%) other baselines in the four metrics, except for creativity on R1. Because R1 directly retrieves human written hyperboles from HYPO-XL and is not strict about the relevance, it has the advantage of being more creative naturally. An example of this is shown in Table 4. Our model achieves a balance between generating hyperbolic output while preserving the content, which indicates the effectiveness of the "overgenerate-and-rank" mechanism. It is also worth noting that in terms of hyperbolicity, MOVER even performs better than human for 16.7% of the test cases. Table 4 shows a case where MOVER is rated higher than human.

Case Study Table 4 shows a group of generated examples from different systems. MOVER changes the phrase "very bad" in the original input to an extreme expression "sheer hell", which captures the sentiment polarity of the original sentence while providing a hyperbolic effect. R1 retrieves a hyperbolic but irrelevant sentence. R3 replaces the word "very" with "richly", which is not coherent to the context, although the word "richly" may introduce some hyperbolic effects. BART just generates a literal sentence, which seems to be a simple paraphrase. Although human reference provides a valid hyperbolic paraphrase, the annotators prefer our version in terms of fluency, hyperbolicity and relevance. Since our system makes fewer edits to the input than the human reference, we are more likely to win in fluency and relevance. Also, the generated hyperbolic span "sheer hell" presents a more extreme exaggeration than "out of the world" according to the human annotators. More examples of the intermediate over-generation results and final generated outputs are shown in Appendix C.

Despite the interesting results, we also observe the following types of errors in the generated outputs: 1) The output is a paraphrase instead of hyperbole: "*My aim is very certain*" $\rightarrow$ "*My aim is very <u>clear</u>*". 2) The degree of exaggeration is not enough: "*The news has been exaggerated*" $\rightarrow$ "*The news has been <u>greatly</u> exaggerated*". 3) The output is not meaningful: "*I'd love to hang out every day*" $\rightarrow$ "*I'd love to <u>live</u> every day*". We believe that incorporating more commonsense knowledge and generating freeform hyperboles beyond word-level or phrase-level substitutions are promising for future improvement.

Ablation Study We investigate the impact of removing partial or all information during the ranking stage. Results are shown in Table 2. Specifically, if we rank multiple generated outputs by only hyperbole score (w/o para score), or randomly select one as the output (w/o hypo ranker), the performance will become worse. Note that we do not report the ablation result for ranking only by paraphrase score (w/o hypo score), because it has the same problem with COPY: a generated sentence that directly copies the input will result in the highest paraphrase score and thus be selected as the final output.

Furthermore, we note that the experiments on R3 and BART also serve as ablation studies for the text infilling model in Section 3.2 as they substitute the fine-tuned BART with a retrieve-and-replace method and a non-fine-tuned BART, respectively.

5 Related Work

Hyperbole Corpus Troiano et al. (2018) built the HYPO dataset consisting of 709 hyperbolic sentences with human-written paraphrases and lexically overlapping non-hyperbolic counterparts. Kong et al. (2020) also built a Chinese hyperbole dataset with 2680 hyperboles. Our HYPO-L and HYPO-XL are substantially larger than HYPO and we hope they can facilitate computational research on hyperbole detection and generation.

Figurative Language Generation As a figure of speech, hyperbole generation is related to the general task of figurative language generation. Previous studies have tackled the generation of metaphor (Yu and Wan, 2019; Stowe et al., 2020; Chakrabarty et al., 2021; Stowe et al., 2021), simile (Chakrabarty et al., 2020c; Zhang et al., 2021), idiom (Zhou et al., 2021), pun (Yu et al., 2018; Luo et al., 2019b; He et al., 2019; Yu et al., 2020), sarcasm (Chakrabarty et al., 2020b), and irony (Zhu et al., 2019). HypoGen (Tian et al., 2021) is a concurrent work with ours on hyperbole generation. However, we share a different point of view and the two methods are not directly comparable. They tackle the generation of *clause-level* hyperboles and frame it as a sentence *completion* task, while we focus on *word-level* or *phrase-level* ones and frame it as a sentence *editing* task. In addition, their collected hyperboles and generated outputs are limited to the "*so...that*" pattern while we do not posit constraints on sentence patterns.

Unsupervised Text Style Transfer Recent advances on unsupervised text style transfer (Hu et al., 2017; Subramanian et al., 2018; Luo et al., 2019a; Zeng et al., 2020) focus on transferring from one text attribute to another without parallel data. Jin et al. (2020) classify existing methods into three main branches: *disentanglement*, *prototype editing*, and *pseudo-parallel corpus construction*. We argue that hyperbole generation is different from text style transfer. First, it is unclear whether "literal" and "hyperbolic" can be treated as "styles", especially the former one. Because "literal" sentences do not have any specific characteristics at all, there are no attribute markers (Li et al., 2018) in the input sentences, and thus many text style transfer methods based on *prototype editing* cannot work. Second, the hyperbolic span can be lexically separable from, yet strongly dependent on, the context (Section 3.1). On the contrary, *disentanglement-based* approaches for text style transfer aim to separate content and style via latent representation learning. Third, we would like to point out that MOVER could also be used for *constructing pseudo-parallel corpus* of literal-hyperbole pairs given enough literal sentences as inputs, which is beyond the scope of this work.

Unsupervised Paraphrase Generation Unsupervised paraphrase generation models (Wieting et al., 2017; Zhang et al., 2019b; Roy and Grangier, 2019; Huang and Chang, 2021) do not require paraphrase pairs for training. Although hyperbole generation also needs content preservation and lacks parallel training data, it is still different from paraphrase generation because we need to create a balance between paraphrasing and exaggerating. We further note that the task of metaphor generation (Chakrabarty et al., 2021), which replaces a verb (e.g., "*The scream filled the night*" $\rightarrow$ "*The scream* <u>*pierced*</u> *the night*"), is also independent of paraphrase generation.

6 Conclusion and Future Work

We tackle the challenging task of figurative language generation: hyperbole generation from literal sentences. We build the first large-scale hyperbole corpus HYPO-XL and propose an unsupervised approach MOVER for generating hyperbole in a controllable way. Automatic and human evaluation results show that our model is successful at generating hyperbole. The proposed generation pipeline has better interpretability and flexibility compared to potential end-to-end methods. In future, we plan to apply our "mask-overgenerate-rank" approach to the generation of other figurative languages, such as metaphor and irony.

7 Ethical Consideration

The HYPO-XL dataset is collected from a public website Sentencedict.com and we have asked for the website owners' permission for using their data for research purposes. It does not contain any explicit detail that leaks a user's personal information including name, health, racial or ethnic origin, religious or philosophical affiliation or beliefs, sexual orientation, etc.

Our proposed method MOVER is based on the pretrained language model, which is known to capture the bias reflected in the training data. Note that MOVER might be used for malicious purposes because it does not have a filtering mechanism that checks the toxicity, bias, or offensiveness of input sentences. Therefore, MOVER could potentially generate harmful or biased content that may offend certain groups or individuals. We suggest interested parties carefully check the generated content and examine the potential biases before deploying MOVER in real-world applications.

References

Tazin Afrin and Diane Litman. 2018. Annotation and classification of sentence-level revision improvement. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 240–246, New Orleans, Louisiana. Association for Computational Linguistics.

Tuhin Chakrabarty, Debanjan Ghosh, Smaranda Muresan, and Nanyun Peng. 2020a. Rˆ3: Reverse, retrieve, and rank for sarcasm generation with commonsense knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7976–7986, Online. Association for Computational Linguistics.

Tuhin Chakrabarty, Debanjan Ghosh, Smaranda Muresan, and Nanyun Peng. 2020b. R^3: Reverse, retrieve, and rank for sarcasm generation with commonsense knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7976–7986.

Tuhin Chakrabarty, Smaranda Muresan, and Nanyun Peng. 2020c. 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.

Claudia Claridge. 2010. *Hyperbole in English: A corpus-based study of exaggeration*. 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.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

He He, Nanyun Peng, and Percy Liang. 2019. Pun generation with surprise. 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 1734–1744.

Michael Heilman and Noah A Smith. 2009. Question generation via overgenerating transformations and ranking. Technical report, Carnegie-Mellon Univ Pittsburgh pa language technologies insT.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International Conference on Machine Learning*, pages 1587–1596. PMLR.

Kuan-Hao Huang and Kai-Wei Chang. 2021. Generating syntactically controlled paraphrases without using annotated parallel pairs. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1022–1033.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2020. Deep learning for text style transfer: A survey. *arXiv preprint arXiv:2011.00416*.

Li Kong, Chuanyi Li, Jidong Ge, Bin Luo, and Vincent Ng. 2020. Identifying exaggerated language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7024–7034, Online. Association for Computational Linguistics.

Roger J. Kreuz and Richard M. Roberts. 1993. The empirical study of figurative language in literature. *Poetics*, 22(1):151–169.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 737–762, Online. Association for Computational Linguistics.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

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.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Sun. 2019a. A dual reinforcement learning framework for unsupervised text style transfer. *arXiv preprint arXiv:1905.10060*.

Fuli Luo, Shunyao Li, Pengcheng Yang, Lei Li, Baobao Chang, Zhifang Sui, and Xu Sun. 2019b. Pun-gan: Generative adversarial network for pun 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 3388–3393.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Nikola I Nikolov, Eric Malmi, Curtis Northcutt, and Loreto Parisi. 2020. Rapformer: Conditional rap lyrics generation with denoising autoencoders. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 360–373.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark. 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.

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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

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*. Association for Computational Linguistics.

Aurko Roy and David Grangier. 2019. Unsupervised paraphrasing without translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6033–6039.

Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. 2019. Long and diverse text generation with planning-based hierarchical variational model. 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 3257–3268, Hong Kong, China. Association for Computational Linguistics.

Kevin Stowe, Tuhin Chakrabarty, Nanyun Peng, Smaranda Muresan, and Iryna Gurevych. 2021. Metaphor generation with conceptual mappings. *arXiv preprint arXiv:2106.01228*.

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

Sandeep Subramanian, Guillaume Lample, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2018. Multiple-attribute text style transfer. *arXiv preprint arXiv:1811.00552*.

Chenhao Tan and Lillian Lee. 2014. A corpus of sentence-level revisions in academic writing: A step towards understanding statement strength in communication. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 403–408, Baltimore, Maryland. Association for Computational Linguistics.

Yufei Tian, Arvind krishna Sridhar, and Nanyun Peng. 2021. HypoGen: Hyperbole generation with commonsense and counterfactual knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1583–1593, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Enrica Troiano, Carlo Strapparava, Gözde Özbal, and Serra Sinem Tekiroğlu. 2018. A computational exploration of exaggeration. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3296–3304, Brussels, Belgium. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

John Wieting, Jonathan Mallinson, and Kevin Gimpel. 2017. Learning paraphrastic sentence embeddings from back-translated bitext. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 274–285.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhiwei Yu, Jiwei Tan, and Xiaojun Wan. 2018. A neural approach to pun generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1660.

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, Volume 1 (Long and Short Papers)*, pages 861–871, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhiwei Yu, Hongyu Zang, and Xiaojun Wan. 2020. Homophonic pun generation with lexically constrained rewriting. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2870–2876.

Kuo-Hao Zeng, Mohammad Shoeybi, and Ming-Yu Liu. 2020. Style example-guided text generation using generative adversarial transformers. *arXiv preprint arXiv:2003.00674*.

Jiayi Zhang, Zhi Cui, Xiaoqiang Xia, Yalong Guo, Yanran Li, Chen Wei, and Jianwei Cui. 2021. Writing polishment with simile: Task, dataset and a neural approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14383–14392.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019a. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Xinyuan Zhang, Yi Yang, Siyang Yuan, Dinghan Shen, and Lawrence Carin. 2019b. Syntax-infused variational autoencoder for text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2069–2078.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019c. Paws: Paraphrase adversaries from word scrambling. *arXiv preprint arXiv:1904.01130*.

Jianing Zhou, Hongyu Gong, Srihari Nanniyur, and Suma Bhat. 2021. From solving a problem boldly to cutting the gordian knot: Idiomatic text generation. *arXiv preprint arXiv:2104.06541*.

Mengdi Zhu, Zhiwei Yu, and Xiaojun Wan. 2019. A neural approach to irony generation. *arXiv preprint arXiv:1909.06200*.

Dataset	# Hypo.	# Non.	# Para.	# Total
HYPO	709	698	709	2,116
HYPO-L	1,007	2,219	-	3,226
HYPO-XL	17,862	-	-	17,862

Table 5: Comparision of different hyperbole datasets (corpora) in terms of hyperbolic (Hypo.), non-hyperbolic (Non.) and paraphrase (Para.) sentences.

POS	#	Hyperbole Example
NN	19	His words confirmed *everything*.
RB	15	He descanted *endlessly* upon the wonders of his trip.
JJ	14	Youth means *limitless* possibilities.

Table 6: Three most common POS n-grams of hyperbolic spans in 94 randomly sampled hyperboles from HYPO-XL. Hyperbolic spans are in *italic*.

A Additional Dataset Statistics

We provide a brief comparison of HYPO, HYPO-L and HYPO-XL (Section 2) in Table 5 to further clarify the data collection process.

We also annotate the hyperbolic spans (Section 3.1) for the 94 real hyperboles in Section 2.3 and show some examples of the most common POS n-grams of hyperbolic spans in Table 6. We further follow Troiano et al. (2018) to annotate the types of exaggeration along three dimensions: "measurable", "possible" and "conventional". A hyperbole is "measurable" if it exaggerates something which is objective and quantifiable. A hyperbole is rated as "possible" if it denotes an extreme but conceivable situation. A hyperbole is judged as "conventional" if it does not express an idea in a creative way. However, we note that there are no absolute answers for these three questions and the annotation results may be subjective. Each hyperbole is either YES or NO for each dimension and the reported numbers in Table 7 are for YES.

B An Illustration of the Unexpectedness Score

Figure 2 illustrates the cosine distance of word pairs in the sentence "*I've drowned myself trying to help you*". The words in the span "*drowned myself*" are distant from other words in terms of word embedding similarity.

Type	#	Hyperbole Example
Measurable	44	At any moment, I feared, the boys could *snap my body in half* with just one concerted shove.
Possible	27	The words caused a shiver to *run a fine sharp line* through her.
Conventional	65	She is *forever* picking at the child.

Table 7: Three types of exaggeration in 94 randomly sampled hyperboles from HYPO-XL. Hyperbolic spans are in *italic*.

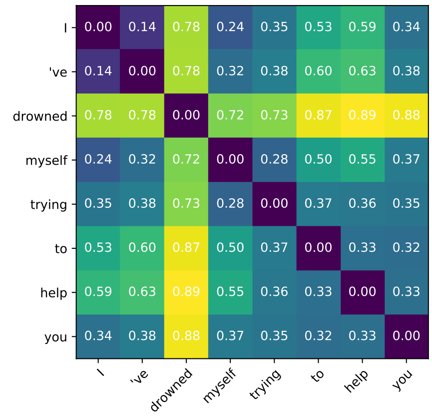

Figure 2: A visualization of the cosine distance matrix of the hyperbolic sentence "*I've drowned myself trying to help you*".

C More Generated Examples

Table 8 shows the over-generation results for a literal input, with their hyperbole and paraphrase scores. On the one hand, our system can generate different hyperbolic versions, like the generated words "cannot", "unyielding", and "alive". This is reasonable since there might be multiple hyperbolic paraphrases for a single sentence. It is only for comparison with other baselines that we have to use the ranker to keep only one output, which inevitably undermines the strength of our approach. On the other hand, our ranker filters out the sentence if the infilling text violates the original meaning, which can be seen from the last row of Table 8. In this way, we gain explicit control over hyperbolicity and relevance through a scoring function, and endow MOVER with more explainability.

Table 9 shows more examples of generated outputs from different systems and human references.

Generated Hyperbole s	$hypo(s)$	$para(s)$	$score(s)$
You have ravished me away by a power I *cannot* resist.	0.962	0.954	0.962
You have ravished me away by a power I find *unyielding* to resist.	0.960	0.959	0.960
You have ravished me *alive* by a power I find difficult to resist.	0.954	0.931	0.954
You have *driven* me away by a power I find difficult to resist.	0.858	0.914	0.858
You have ravished me away *with a beauty* I find difficult to resist.	0.958	0.778	0.000

Table 8: Intermediate results of the input literal sentence "*You have ravished me away by a power I find difficult to resist*" after the over-generation steps (Section 3.2). Their ranking scores (Section 3.3) are displayed in the second to the fourth columns. Generated hyperbolic text spans are in *italic*.

System	Sentence	Flu.	Hypo.	Crea.	Rel.
LITERAL	At that point, the presidency was hard to recover.	-	-	-	-
MOVER	At that point the presidency was *virtually impossible* to recover.	-	-	-	-
R1	*The destruction of a President with its collapse of executive authority was too staggering to contemplate.*	**W**	**W**	T	**W**
R3	At that point the presidency was *staggering* to recover.	**W**	**W**	L	**W**
BART	At that point the presidency was *too fragile* to recover	T	**W**	T	T
HUMAN	At that point, the presidency was *fatally wounded*.	T	**W**	**W**	**W**
LITERAL	His piano playing is very bad.	-	-	-	-
MOVER	His piano playing is *beyond* bad.	-	-	-	-
R1	Her piano playing is *absolute magic*.	T	T	L	**W**
R3	His piano *guitar* is very bad.	T	T	L	L
BART	His piano playing is very *good*.	T	**W**	**W**	**W**
HUMAN	His piano playing is *enough to make Beethoven turn in his grave*.	T	L	L	**W**
LITERAL	The professor humiliated me in front of the class.	-	-	-	-
MOVER	The professor humiliated me in *every conceivable way*.	-	-	-	-
R1	*She infected the whole class with her enthusiasm.*	**W**	**W**	**W**	**W**
R3	*That lecture* humiliated me in front of the class.	T	**W**	T	T
BART	The professor humiliated me *and the rest* of the class.	**W**	**W**	**W**	**W**
HUMAN	The professor *destroyed* me in front of the class.	T	L	**W**	**W**
LITERAL	It annoys me when you only drink half of the soda.	-	-	-	-
MOVER	It *kills* me when you only drink half of the soda.	-	-	-	-
R1	*That was the best ice-cream soda I ever tasted.*	T	**W**	**W**	**W**
R3	It annoys me when you only drink *boredom* of the soda.	T	**W**	**W**	T
BART	It annoys me when you only drink half of *it*.	**W**	**W**	**W**	**W**
HUMAN	It *drives me crazy* when you only drink half of the soda.	T	**W**	T	T

Table 9: Pairwise evaluation results (Win[W], Lose[L], Tie[T]) between MOVER and generated outputs of baseline systems. Changed text spans are in *italic*.