HypoGen: Hyperbole Generation with Commonsense and Counterfactual Knowledge

Yufei Tian, Arvind krishna Sridhar, and Nanyun Peng
Computer Science Department, University of California, Los Angeles
{yufeit, arvindkrishna97, violetpeng}@cs.ucla.edu

Abstract

A hyperbole is an intentional and creative exaggeration not to be taken literally. Despite its ubiquity in daily life, the computational explorations of hyperboles are scarce. In this paper, we tackle the under-explored and challenging task: sentence-level hyperbole generation. We start with a representative syntactic pattern for intensification and systematically study the semantic (commonsense and counterfactual) relationships between each component in such hyperboles. Next, we leverage the COMeT and reverse COMeT models to do commonsense and counterfactual inference. We then generate multiple hyperbole candidates based on our findings from the pattern, and train neural classifiers to rank and select high-quality hyperboles. Automatic and human evaluations show that our generation method is able to generate hyperboles creatively with high success rate and intensity scores.

1 Introduction

Hyperboles invoke the use of exaggeration as a rhetorical device or figure of speech. It is interactive, amusing, and is the second most common among all tropes of figurative language, only after metaphors (Kreuz and MacNealy, 1996). By definition, a hyperbolic expression exceeds the credible limits of fact in the given context, whereas a literal expression agrees with the extralinguistic facts in the given context (Claridge, 2010). For example in Figure 1, *"The party is so lit even the wardrobe is dancing!"* is considered as a hyperbole because making a lifeless object to dance is impossible; it is an intentional and creative way of exaggerating how lit the party is, and is not meant to be taken literally. In contrast, *"The party is so lit (that) even my introvert friend has a good time!"* is considered literal, because letting introvert people have a good time is realistic and hence not an overstatement.

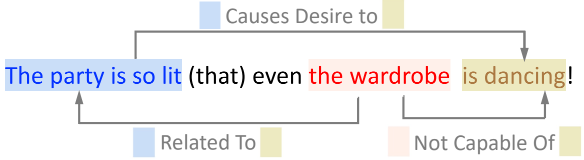

Figure 1: An illustration of the commonsense and counterfactual relationships within a *clause or sentence level* hyperbole. The input prompt (**A**), subject in the clause (**B**), predicate in the clause (**C**), and the relationships between them are colored in blue, red, brown and grey. In this example, that *'the party is lit'* causes the desire to *'dance'*. In addition, *'the wardrobe'* is related to *'the party'*, and is not capable of *'dancing'*.

Despite its abundance, identifying and generating hyperboles remain under-explored. Compared to the many efforts on other figurative languages such as puns, sarcasms, metaphors and similes (He et al., 2019; Chakrabarty et al., 2020a; Su et al., 2020; Yu and Wan, 2019; Chakrabarty et al., 2020b), the exploration of hyperboles is still in the infancy stage: NLP researchers have just started to look at automatic hyperbole detection (Troiano et al., 2018; Kong et al., 2020). According to Claridge (2010), hyperboles are divided into two categories: those at the *word or phrase level* and those at the *clause or sentence level*. The former is less creative because it is easily achievable via lexicon substitution (Norrick, 2012). For example, replacing most time durations with 'a millisecond' will make noncreative exaggerations to emphasize something is fast, without needing to understand the context.

In this work, we target at generating the more challenging type of hyperboles, i.e. **clause or sentence level hyperboles**. According to McCarthy and Carter (2004), clause-level hyperboles consist of counterfactuality and syntactic support. Inspired by the linguistic theory that '*so + adj/adv + (that) + a declarative clause*' is a significant pattern with both prototypical syntactic and semantic function as overstatement (Bäcklund, 1973; Lorenz, 2002), we leverage the *so...that* pattern, where 'that' is omittable, as a starting point to analyze and gener-

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 1583–1593
November 7–11, 2021. ©2021 Association for Computational Linguistics

ate hyperboles.

Claridge (2010) state that within the *so...that* pattern, the sentence serves as a result of the prompt (**A**) and that the sentence itself creates impossible worlds. Inspired by this, we systematically investigate the semantic (commonsense or counterfactual) relationships between the components within the *so...that* pattern. Specifically, we partition each sentence into three parts: the literal prompt (**A**), the subject of the clause (**B**), and the predicate of the clause (**C**), as illustrated in Figure 1, and conduct detailed annotation and analysis. We discover that 6 semantic relations among **A**, **B**, and **C** account for over 95% of all hyperboles with the *so...that* pattern. This indicates that if a generation model can cover these 6 relationships, it is able to generate almost all hyperboles with such pattern.

With the annotated relationships as background knowledge, we build a hyperbole generation model that takes a literal prompt (**A**) as input and outputs a hyperbole clause (**B** and **C** combined). To this end, we train a reverse COMeT model to generate commonsense and counterfactual phrases along with the COMeT model (Bosselut et al., 2019), and rank the generated candidates with a hyperbole identifier. Finally, we break the restrictions of the *so...that* pattern, and generate hyperboles with diverse syntactic structures using a syntactically controlled paraphrase model. To the best of our knowledge, we are the first to analyze the relations of the logical components within hyperboles, and the first to automatically generate hyperboles. We summarize our contributions as follow:

- We create an English hyperbole dataset from the online discussion forum, Reddit, and analyze hyperboles in the *so...that* pattern to understand the commonsense and counterfactual relationships between each component within such pattern. Our analysis discover that 6 major relations cover 95% of all occurrences. This provide guidelines for us to design models for automatic hyperbole generation. (Details can be found in Section 3)
- Based on the analysis, we propose **HypoGen**, a hyperbole generation model that takes a literal prompt as input, and generate hyperbole sentences. Automatic and human evaluations show that our best model **HypoGen**$_{Spec}$ is able to generate high-quality hyperboles with high success rate.[1] (Details can be found in Section 4)

Literal	Postgraduate **literally** refers to any degree after an undergraduate degree.
Hyperbole	My boyfriend was so hungry, he **literally** swallowed his plate.
Literal	**I swear** to God I don't know how that cat got there!
Hyperbole	**I swear** to Jeebus I will burn this building to the ground!

Table 1: Examples of retrieved sentences from Reddit that contain keywords 'literally' and 'I swear'. Whether these sentences are hyperbole or literal depends on the semantic meaning, not the existence of such keywords.

- We further propose to apply syntactically controlled paraphrase generation model to break the *so...that* pattern and generate creative hyperboles with diverse syntactic structures.

2 Task Definition

Given an input prompt (**A**), we aim to generate clause or sentence level hyperboles by completing that clause. For example, if the input is *'the party is lit'*, our task is to generate *'the wardrobe'* (a subject **B**) and *'is dancing'* (a predicate **C**) to make the full sentence (*'the party is so lit that even the wardrobe is dancing'*) a hyperbole.

3 Data Collection and Analysis

Section 3.1 introduces how we collect hyperboles and non-hyperboles sentences from Reddit. In Section 3.2, we describe the procedure for a detailed second-round annotation: sensical relationship annotation for hyperboles with the *so...that* pattern.

3.1 Collection of Hyperboles

Considering their ubiquity in people's everyday conversation, we collect hyperboles from online discussion forums. We first crawl thousands of sentences from Reddit that contain different patterns or adverb keywords (phrases) that are potential hyperboles, such as *I swear*, *literally*, and *so...that* (Mora, 2009). Table 1 illustrates how the retrieved sentences containing such keywords can be both hyperboles (positive examples) and literal (negative examples) sentences. Thus, we instruct human annotators to decide if a given sentence is hyperbole or not. In total, 3,300 sentences are annotated and each sentence is annotated by at least three annotators. The worker agreement with aggregate, or "Wawa", which measures the average number of times that the rators' response agree with the aggregate answer , is 0.72.

[1] Our code and data are available at `https://github.com/NinaTian98369/HypoGen`

Rule	Occurrence	A <-> B	B <-> C	A <-> C	Example Sentence
1	29.3%	B -> A.1, Related To	Not Capable Of	Causes	The party is so lit that even the wardrobe is dancing.
2	28.4%	B -> A.2, Has Property or Related To	Not Capable Of	Causes	He is so tall that even a mountain looks up to him.
3	17.7%	B -> A.1, Identical	Not Capable Of	Causes	My boyfriend was so hungry, he even swallowed his plate.
4	4.2%	B -> A.1, Related To	Characteristic Action	Causes	The monster's face is so ugly that Ins refuses to load it.
5	11.2%	B -> A.2, Has Property or Related To	Characteristic Action	Causes	My personality is so dry that a cactus flourishes inside.
6	4.6%	B -> A.1, Identical	Characteristic Action	Causes	That ball is so heavy that it is causing tidal waves.

Table 2: Our annotation results: we identify six sensical relations for the *so...that* pattern. We list the percentage of occurrences, names of relation between **AB**, **AC**, **BC**, and example sentences. Here **A**.1 and **A**.2 stand for the subject of **A** and the head word modified by *so*.

We call our collected data HYPO-Red. HYPO-Red is valuable because both negative and positive samples contain such keywords, meaning that an identification model must avoid the superficial patterns and focus on the crux of hyperbole: 1) going to extreme and 2) counterfactuality not meant to be taken literally. Using our collected data, we trained a hyperbole identifier by finetuning BERT. Details can be found in Section 5.1.

3.2 Relationship Annotation

The *so...that* Pattern We already know that clause-level hyperboles include counterfactuality and syntactic support. Moreover, the content clauses always express the resultant meaning of the prompts (e.g., 'want to dance' is the result of 'the party is lit') and that the clause itself creates impossible worlds (e.g., 'wardrobe is dancing' creates an impossible world) (Claridge, 2010). However, those observations are not concrete enough for a systematical exploration of complicated hyperboles and hyperbole generation. To uncover the underlying sensical (commonsense and counterfactual) relationships of hyperboles, we study the *so...that* pattern because it is both representative and easy to spot using keywords. Specifically, we randomly collected 500 hyperboles that contain either *so...that* and *so...even*, and then partition the pattern into three components: the literal prompt (**A**), the subject in the clause (**B**) and the predicate (verbal phrase) in the clause (**C**). We then instruct six annotators to annotate these 500 hyperboles.

Annotation Procedure We provide the annotators with a few seed options present in linguistic papers (such as **C** as the result of the **A**). The annotators are asked to independently label the relationships within a sentence, i.e., between **AB**, **BC**, and **CA**. All annotators receive detailed instructions about how to react if they find a new sensical relationship or none of the seed options fit. Each sentence is annotated by three people.

Annotation Results We find that 6 sensical relations account for over 95% of all occurrences. We report their percentage of occurrences, names for each relation, and example sentences in Table 2. First, we discover that **C** is always the result of **A**. Next, the interaction of **B** and **C** creates counterfactuality (Claridge, 2010). Either **B** is not capable of conducting the action of **C** (rule 1-3), or **C** is one of **B**'s characteristic actions, but surely unrealistic given the context of **A** (rule 4-6). For instance, for rule 5, 'a cactus' grows in dry area and 'flourish' is one of its characteristic actions. However, a cactus *cannot* grow inside one's mind. Given the context of 'my personality is dry', that 'a cactus flourishes inside' is unrealistic.[2]

Finally, we discover that the literal prompt (**A**) can be further divided into **A**.1: the subject and **A**.2: the head word modified by *so* (usually an adjective or adverb). In total, there are three logical relationships between **AB**: **1) B** is related to **A**.1 (rule 1&4), **2) B** is related to or shares the same attribute with **A**.2 (rule 2&5), and **3) B** is identical to **A**.1(rule 3&6). For example, for *'He is so tall that a mountain looks up to him.'*, 'He' is **A**.1 and 'tall' is **A**.2. Since a mountain (**B**) has the attribute of tall (**A**.2), but is not capable of looking up (**C**), this hyperbole a sample from rule 2.

For all six rules, we use Spearman's correlation

[2] Occasionally, **C** may also be the *inverse* characteristic action of **B**, depending on the context of **A** (see the example sentence of rule 4).

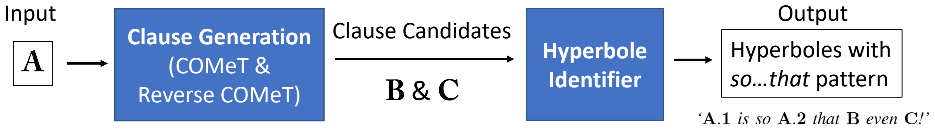
Input
A
Clause Generation (COMeT & Reverse COMeT)
Clause Candidates
B & C
Hyperbole Identifier
Output
Hyperboles with *so...that* pattern
'A.1 *is so* A.2 *that* B *even* C!'

Figure 2: A high-level diagram of our hyperbole generation pipeline: **HypoGen**. We first generate clause candidates with the COMeT and reverse COMeT model, and then rank the candidates with a hyperbole classifier.

to measure the inter-annotator agreement (IAA). The IAA score is 0.88, meaning that the raters have substantially strong agreement. We call the annotated data HYPO-so.

4 Methodology

In this section, we introduce several components for our generation model. In Section 4.1, we introduce the COMeT model (Bosselut et al., 2019) and its reverse model that favors less frequent and more creative outputs. In Section 4.2 we design an algorithm to generate multiple hyperbole candidates. Section 4.3 explores two possible classifiers as hyperbole identifiers to select the best candidates. A diagram is shown in Figure 2. Furthermore, we propose to use paraphrasing techniques to break the pattern restriction and generate hyperboles with diverse syntactic structures in Section 4.4.

4.1 COMeT and Reverse COMeT Model

COMeT and ConceptNet COMET (Bosselut et al., 2019) is a pre-trained generative model finetuned on ConceptNet (Speer et al., 2017), a knowledge graph of commonsense knowledge in the format of <Entity1 (E1), Relation (R), Entity2 (E2)>. We utilize the pretrained COMeT model[3] to generate multiple candidates with E1 and R as inputs. For example, given E1 as 'the party is lit' and R as 'cause desire', COMET predicts E2 as 'to dance'.

Reverse COMeT Model Now that we have the COMeT model to generate diverse commonsense descriptions from left to right, we also need another model to predict E1 from E2 and R. To this end, we train a reverse COMeT model that takes E2 as input, and E1 as output. That is to say, the ordering of the original ConceptNet tuple is reversed with respect to the COMeT model.

On top of this, we add two mechanisms to generate even more creative descriptions. First, the reverse COMeT model favors phrases with novel or less frequent words. During the decoding step, we re-score and rank the generated beams. Inspired by mutual information, the re-ranking function is:

$$\mathcal{R}_{||} = \frac{e^{\frac{P(b_k)}{T}}}{\frac{\sum_{i=1}^{T} P_{b_k}(i)}{T}}, \tag{1}$$

where $P(b_k)$ is the probability of generation beam k, T is the length of beam k, $P_{b_k}(i)$ is the unigram probability of the i^{th} word in beam k and $\frac{\sum_{i=1}^{T} P_{b_k}(i)}{T}$ is the unigram probability that the beam exists in the corpora.

Retrieved Simile	Created Triplet
as impertinent as the drama	<drama, HP, impertinent>
as silent as the grave	<grave, HP, silent>
as pale as a sheet	<sheet, HP, pale>
as effortless as breathing	<breathing, HP, effortless>

Table 3: Examples of the similes we retrieved, and the triplets we created in the format of: <Entity1, Has Property (HP), Entity2)>.

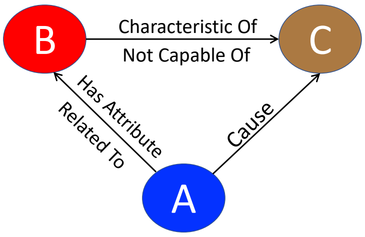

Figure 3: An illustration of the generation flow.

Second, we augment the original training triplets in the ConceptNet data(Speer et al., 2017) with figurative samples retrieved from the simile corpora (Chakrabarty et al., 2020b). Table 3 shows a few examples for the original similes and their relationships. For instance, we map the simile *'as impertinent as the drama'*, to <drama, HasProperty, impertinent>.[4]

4.2 Clause Candidate Generation

Since counterfactuality is a salient component of successful hyperboles, language models view hyperboles as less predictable than literals (Troiano et al., 2018). Therefore, instead of generating the

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

[4] The total number of additional figurative triplets for training the reverse COMeT model is 13,640.

Algorithm 1 Hyperbole Clause Generation

```
function GenHyper($A$)
Input: Input prompt $\mathbf{A}$
Output: List of candidate <B, C> pairs $cand$
    Initialize $Bs$, $cand$ to empty list
    $subject$, $head\_word$ = parse($\mathbf{A}$)
    $Bs$ += getPreds($subject$, 'RelatedTo')
    $Bs$ += getPreds($head\_word$, 'HasProperty')
    $Bs$ += subject
    for B in $Bs$ do
        for C in getPreds(A, 'Causal') do
            $cand$.append(<B,C>)
        end for
        for C in getPreds(B, 'CharacteristicOf') do
            $cand$.append(<B,C>)
        end for
    end forreturn $cand$        ▷ Fit into the so...that pattern.
end function
```

clause fully, we **separately** generate the clause's subject ($\mathbf{B}$) and predicate ($\mathbf{C}$). Our generation flow is illustrated in Figure 3 and Algorithm 1.

Generating B from A We first parse the input prompt ($\mathbf{A}$) into the subject ($\mathbf{A}.1$) and the headword ($\mathbf{A}.\mathbf{2}$). We then generate $\mathbf{B}$ using the `RelatedTo` and `HasProperty` with the COMeT and reverse COMeT model. Following the COMeT paper (Bosselut et al., 2019), we also compute the conditional log-likelihood of predicting the object tokens X:

$$\mathcal{L} = -\sum_{t=|e1|+|r|}^{|e1|+|r|+|e2|} \log P\left(x_t \mid x_{<t}\right), \qquad (2)$$

where $|e1|$, $|r|$, and $|e2|$ are the number of tokens in e1, relation, and e2, respectively. We denote the likelihood $\mathcal{L}$ as l_{AB} when the likelihood is calculated from generating $\mathbf{B}$ from $\mathbf{A}$.

Generating C from A and from B There are two ways to generate $\mathbf{C}$: from $\mathbf{A}$ and from $\mathbf{B}$. Given $\mathbf{A}$, we can leverage several causal relationships, such as `CauseDesire`, `Causes`, and `HasSubevent`. Given $\mathbf{B}$, we produce i) predicates that $\mathbf{B}$ is not capable of, using `NotCapableOf` directly available in ConceptNet; and ii) characteristic actions of $\mathbf{B}$, from the following relationships `DefinedAs`, `CapableOf`, `IsA`, and `UsedFor`. We also compute the conditional log-likelihoods and call them l_{AC} and l_{BC}.

Finally, we assemble pieces of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ into the 'so...that' pattern. The candidate sentence is: '$\mathbf{A}.\mathbf{1}$ *is so* $\mathbf{A}.\mathbf{2}$ *that* $\mathbf{B}$ *even* $\mathbf{C}$*!*'.

Grammar Error Correction When we assemble pieces of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ into the 'so...that' pattern, such manipulation can cause certain grammar errors such as mismatch of verb tenses, or singularity/plurality. While writing a rule-based grammar error correction (GEC) algorithm can be effective for a set of these common errors, we hope to fix open-ended grammar errors. Therefore, we choose the GEC model by Zhao et al. (2019), a widely used neural architecture for the GEC problem with copy-augmented architecture and token-level and sentence-level multi-task learning.

4.3 Hyperbole Candidate Ranking

We build two classifiers to score and rank the hyperbole candidates. We later compare their performance through human evaluation and ablation in Section 6 and Section 7.

The Generic Classifier First, we train a generic hyperbole classification model by finetuning BERT (Devlin et al., 2018) with the data collected in Section 3.1. Before training, we deliberately remove all the keywords such as *I swear*, *literally*, *so ... that* to eliminate the influence of superficial cues. We call the model Clf_G and predicted probability p_G.We call the generation method with Clf_G as classifier **HypoGen**$_{Gene}$.

The Specific Classifier The second classifier is specifically designed for hyperboles with the *so...that* pattern. We posit that values of l_{AB}, l_{AC}, and l_{BC} indicate the intensity of a hyperbole when Clf_G is not fully reliable. Hence, we compute the values of p_G, l_{AB}, l_{AC} and l_{BC} for 600 *so...that* sentences (half of them are hyperboles and half are literals), and then train a multiple layer perceptron with these four variables as input features. We call the model Clf_S and predicted probability p_S:

$$p_S = \mathbf{MLP}(p_G, l_{AB}, l_{AC}, l_{BC}) \qquad (3)$$

Note that to avoid information leakage, the training data for Clf_G and Clf_S do not overlap. We call the generation method with Clf_S as classifier **HypoGen**$_{Spec}$.

4.4 Breaking the *so...that* Pattern

So far we have managed to generate hyperboles with the *so...that* pattern. As an extension to our proposed **HypoGen**, we posit that a paraphrasing module is helpful to break such pattern and hence generate hyperboles with diverse syntactic structures. Specifically, we use the syntactically-controlled paraphrasing model by Sun et al. (2021) as an off-the-shelf tool, because it achieves state-of-the-art performances on semantic preservation and syntactic conformation. It leverages pretrained BART (Lewis et al., 2019) and adds deliberately chosen syntactical control via a retrieval-based selection module to generate fluent paraphrases.

We use **HypoPara** to denote **HypoGen**$_{Spec}$ added by such a paraphrasing model.

5 Experiments

5.1 Hyperbole Detection Model

Clf_G**.** Recall that to further remove the influence of superficial clues for hyperboles, we delete all keywords used to crawl hyperboles from Reddit. Next, we balance the training data and then finetune the BERT-base model (Devlin et al., 2018) to train a binary classification model. We also compare our classification model with that of Troiano et al. (2018) by testing on their dataset, HYPO-en.

Clf_S**.** We train a simple MLP for Clf_S and use grid search to find the best hyper-parameters. The best neural network has 2 hidden layers with sizes of 8 and 4. Alpha is 1×10^{-4} for regularization.

5.2 Baselines

Sim Retrieval We first try a naive phrase matching model where we retrieve sentences that contain the input prompt (**A**). However, the success rate of exact match is only 3%, so we utilized a less stringent matching function called Sim Retrieval. Sim Retrieval uses cosine similarity of token embeddings to find the sentence that is semantically similar to a input prompt (**A**). For both retrieval based baselines, we retrieve from news commentaries dataset from 2007 to 2020 [5] because the corpus is large and is likely to contain hyperboles.

Fine-tuned BART We finetune the model with the input prompts (**A**) as input to the encoder and the full hyperboles as the output by the decoder.

Ablations of HypoGen To study the role of each model component, we compare four variations of our main model. We rank the generated hyperbole candidates with 1) p_G (**HypoGen**$_{Gene}$), 2) p_S (**HypoGen**$_{Spec}$), 3) p_G and l_{AC} (we call **HypoGen**$_{Spec}$ w/o **B**), 4) p_G and l_{AB} (we call **HypoGen**$_{Spec}$ w/o **C**).

5.3 Evaluation

Automatic Evaluation For creative generation, it is uncommon to have significant n-gram overlap between the machine-generated and the gold-standard sentences. Therefore, instead of BLEU, we use BERTScore (Zhang et al., 2019) to measure the semantic similarity between machine outputs and human-written hyperboles. In addition, Troiano et al. (2018) propose *unexpectedness* to assess the quality of hyperboles, which refers to the fact that hyperboles are less predictable expressions than literals both for humans and language models. We follow their procedure and compute the sentence *expectedness* as its average token probability predicted by GPT2-large (Radford et al., 2019).

Model	P	R	F-1
Clf_G	0.84	0.83	0.84
Hype-Par (Troiano et al., 2018)	0.76	0.76	0.76

Table 4: Performance of Clf_G and the baseline model Hype-Par on the HYPO-en testset (Troiano et al., 2018)

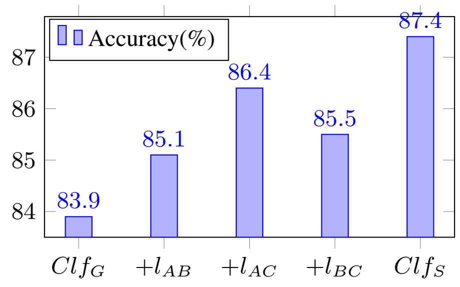

Figure 4: Performance of two hyperbole classifiers (Clf_G and Clf_S) on *so...that* patterns. We also show ablations of each variable: l_{AB}, l_{AC} and l_{BC}.

Human Evaluation Currently available automatic metrics cannot fully reflect the quality of generated hyperboles. Hence, we also conduct human-based evaluation. We first ask the annotators to evaluate if a given sentence is hyperbole, and compute the success rate of each generation model. We then ask a set of 5 criteria to evaluate the generated output: **1)** Intensity of the hyperbole: extent of the exaggeration, **2)** Coherency of the hyperbole: how well the clause is reasonably, meaningfully and understandingly related to the prompt, **3)** Funniness, **4)** Creativity and novelty, and **5)** Grammaticality. Each generation is annotated by four human annotators. They are asked to score each criteria on a scale from 1 (not at all) to 5 (extremely). We evaluate 120 sentences for the gold standard (human) model and each baseline.

[5] http://data.statmt.org/news-crawl/en/

Model		Success Rate	Intensity	Coherency	Funniness	Creativity	Grammar
Human		**84.2%**	**3.50**	**3.41**	**2.81**	**3.05**	**3.82**
Baselines	Naive Retrieval	3.0%	/	/	/	/	/
	Sim Retrieve	28.9%	2.51	2.56	2.14	2.36	2.78
	Fine-tuned BART	44.6%	2.65	2.78	2.23	2.71	$3.27^\dagger$
Proposed	**HypoGen**$_{Gene}$	64.1%	3.03	$2.89^\dagger$	2.46	2.84	3.20
	HypoGen$_{Spec}$ w/o **B**	65.2%	3.12	2.86	2.44	2.80	3.19
	HypoGen$_{Spec}$ w/o **C**	66.3%	3.19	2.79	2.50	2.89	3.12
	HypoGen$_{Spec}$	$67.8\%^\dagger$	$3.23^\dagger$	2.85	$2.54^\dagger$	$2.98^\dagger$	3.13
	HypoPara	48.0%	3.17	2.81	2.40	2.75	3.17

Table 5: Human evaluation results on the success rate and five criteria of hyperbole quality: intensity, coherency, funniness, creativity or novelty, and grammarcality. Boldface in black denotes the human performance; underscore with † denotes the best performance among models.

Model	BERTScore			Expect-edness
	P	R	F1	
Human	**1.00**	**1.00**	**1.00**	**0.095**
Sim Retrieval	0.19	0.28	0.23	0.139
Fine-tuned BART	0.24	0.29	0.27	0.115
HypoGen$_{Gene}$	0.31	0.29	0.30	0.087
HypoGen$_{Spec}$ w/o B	0.31	$0.29^\dagger$	$0.30^\dagger$	0.084
HypoGen$_{Spec}$ w/o C	0.29	0.27	0.28	0.083
HypoGen$_{Spec}$	$0.31^\dagger$	0.29	0.30	$0.083^\dagger$
HypoPara	0.30	0.27	0.28	0.093

Table 6: Automatic evaluation results of our model **HypoGen** and baselines. We report the precision, recall and F1 of BERTScore (higher is better), and expectedness (lower is better). Boldface in black denotes the human performance; underscore with † denotes the best performance among models.

6 Results

6.1 Performance of the Classification Model

The Generic Classifier. Table 4 shows the accuracy of Clf_G and the previous SOTA Hype-Par (Troiano et al., 2018) that uses Skip-Gram representations and several manually defined features. Even though Clf_G is trained on HYPO-Red and tested on HYPO-en (Hype-Par is trained and tested on the same dataset, HYPO-en), our Clf_G still outperforms Hyper-Par by 8%. Tested on HYPO-Red, Clf_G achieves a score of 83.35%. We cannot see how well Hype-Par does on HYPO-Red, because Hype-Par requires computing hand-crafted features on the training data, which is not publicly available.

The Specific Classifier. Figure 4 reports the performances of Clf_G and Clf_S on the task of identifying hyperboles containing *so...that* patterns. Clf_G alone already achieves satisfactory accuracy (83.9%), and Clf_S is 3.5% better than Clf_G. With the addition of l_{AB}, l_{AC} or l_{BC}, model performances have increased by 1.2%, 2.5%, or 1.6%. Among them, the causal relation between A and C contributes most.

6.2 Evaluation Results

We report the results of human and automatic evaluation in Table 5 and Table 6.

Automatic Evaluation Table 6 shows the precision, recall, and F1 of BERTScore and the *expectedness* value of our systems and the baselines. Compared with the baselines, **HypoGen**$_{Spec}$ achieves high BERTScore, meaning that the generations of **HypoGen**$_{Spec}$ are semantically similar to human-written hyperboles. For *expectedness* scores, the retrieval method and fine-tuned BART tend generate more 'normal' and predictable outputs than our systems. However, **HypoGen** is even less predictable than human-written hyperboles. A possible explanation is that human-written ones are both coherent and exaggerating, containing more conjunction words (e.g., the, and, so, that) which contribute to the average word probability.

Human Evaluation Table 5 reports the scores of the five human-evaluated criteria for our model and its variable, human written hyperboles, and the baseline models. To better understand the merits of our proposed model, we also provide four examples of the generated outputs in Table 7. It is interesting that **HypoGen**$_{Spec}$ is annotated to achieve creativity close to that of humans. We attribute such a high creativity score to the counterfactuality introduced in Section 4.2.

For all automatic generation methods, **HypoGen**$_{Spec}$ has the highest success rate (67.8%), intensity of hyperbole (3.23/5), funniness (2.54/5) and creativity (2.98/5). On the other hand, the BART model is the best at producing grammatically correct outputs. Even with the grammar-error-correction model provided by Zhao et al. (2019), **HypoGen** still suffers from grammar errors.

System	Generation	Int	Coh	Fun	Cre
Human	When the party is so lit even the wardrobe is dancing!	**4.50**	**4.50**	**3.75**	**4.25**
Sim Retrieval	The party is so lit that happened after was crazy I thought I was gonna die!	3.25	2.75	2.00	2.00
BART	The party is so lit that even the bugs had too give it a yelp review!	3.75	2.75	2.50	3.50
HypoGen 1	The party is so lit that even the street wants to have fun!	3.75	3.25	2.50	2.75
HypoGen 2	The party is so lit that even the city gets drunk!	4.00	3.75	3.00	3.00
HypoPara 1	What a lit party that the street wants to have fun with it!	3.50	2.75	2.25	2.75
HypoPara 2	Why is party so lit that the city is drunk?	4.00	3.50	2.50	3.00
Human	His drawing is so bright that I cannot open my eye!	4.50	4.75	2.75	4.50
Sim Retrieval	His drawing is so bright, at first its discoverers thought something was wrong,	2.25	3.00	2.75	4.25
BART	His drawing is so bright even god gave up with you before giving him a chin.	3.25	2.50	2.75	3.00
HypoGen 1	His drawing is so bright that even sun adjusts focus!	**5.00**	4.75	4.25	4.00
HypoGen 2	His drawing is so bright that even stars fade away!	4.75	**5.00**	**4.50**	**4.75**
HypoPara 1	How can I learn about such a bright drawing when the sun adjusts the focus?	2.50	2.75	2.25	2.50
HypoPara 2	I 'm sure his picture's so bright that the stars are gone.	4.00	4.00	3.50	3.75
Human	Your forehead is so big even a 787 can land on it.	**5.00**	**4.75**	**4.75**	**4.75**
Sim Retrieval	Your forehead is so big that ordinarily would have threatened to ruin a perfect day for watching TV.	/	/	/	/
BART	Your forehead is so big that even your hairline is running away from it.	4.25	**4.75**	4.00	4.00
HypoGen 1	Your forehead is so big even Eiffel Tower can not fit inside of your head	4.25	3.25	3.75	3.75
HypoGen 2	Your forehead is so big even universe wants to inhabit!	4.75	3.25	4.00	4.25
HypoPara 1	Does eiffel tower fit in your head?	3.75	3.00	3.50	3.75
HypoPara 2	You have such a big forehead that even the universe would want to inhabit it.	4.50	4.50	4.00	4.00
Human	The young artist is so productive, even paintings get moved and start to paint themselves!	**4.25**	**4.75**	**4.00**	**3.75**
Sim Retrieval	The young artist is so productive that age and I didn't make the same mistakes because I was able to learn from her's.	/	/	/	/
BART	The young artist is so productive that even Shia Labeouf tells you not to do it.	3.00	2.25	3.00	2.50
HypoGen 1	The young artist is so productive that Botticelli removes paint from his wall!	4.00	3.00	2.75	3.25
HypoGen 2	The young artist is so productive that Botticelli wants to retire!	3.75	3.50	2.75	2.75
HypoPara 1	Will give rise to the art of youth and even stop selling Botticelli's paintings!	3.75	3.25	2.25	2.50
HypoPara 2	What is the success of young artists for letting Botticelli retire?	2.75	2.75	2.00	2.75

Table 7: Examples of generated outputs from human and different models, and their intensity, coherency, funniness, and creativity scores. We show average scores (over four annotators) on a 1-5 scale, with 1 denoting the worst and 5 the best. The boldface numbers denote the best scores, and underlined numbers denote the second best scores. HypoGen 1 and HypoGen 2 represent two hyperboles generated by **HypoGen**$_{Spec}$

6.3 Breaking the *so...that* Pattern

Based on the evaluation results in Table 5 and the examples in Table 7, it is clear that we are able to generate hyperboles with diverse syntactic structures through paraphrasing. However, the success rate and quality of hyperboles become lower. We believe that since **HypoGen** and **HypoPara** each has its own benefits, a trade-off between diversity and intensity is inevitable. Moreover, since we leverage off-the-shelf paraphrasing models, we believe the performance of **HypoPara** will improve with the development of paraphrasing techniques.

7 Role of Each Component

Here we analyze the role of **A**, **B**, and **C** in **HypoGen**. Ablations of our own models are colored in the grey background in Table 5. First, we discover that **HypoGen**$_{Gene}$ is better at selecting coherent and grammar correct generations then **HypoGen**$_{Spec}$. A possible explanation is that **HypoGen**$_{Gene}$ is finetuned on BERT, and that pretrained language models are good at selecting coherent text. However, **HypoGen**$_{Spec}$ is still considered the best model, because it has the highest success rate and generate the most exaggerated, fun, and creative hyperboles.

Second, compared with the predicate (**C**), the subject of clause (**B**) contributes more to the funniness score and creativity score. We posit that the interplay between **A** and **B** (and also between **B** and **C**) is the dominant factor of novelty, funniness and creativity. Similarly, the predicate (**C**) which is responsible as a result of input, contributes more to the coherency score. We hence posit that the interplay between **A** and **C** determines how well our generation is reasonable and understood.

8 Related Work

8.1 Linguistic Studies on Hyperboles

Our generation model is partially inspired and backboned by various linguistic studies about hyperboles. Claridge (2010) classify hyperboles into word/phrase level and clause/sentence level. The former can be easily achieved via lexicon substitution (Norrick, 2012), while the latter requires more sophisticated world knowledge and hence is more creative, interactive and challenging.

McCarthy and Carter (2004); Mora (2009); Claridge (2010) identify hyperbole as the creation of impossible worlds, unchallenged counterfactuality and syntactic support. Kunneman et al. (2015) focus on the presence of language intensity as a potential cue to hyperbole. (Bäcklund, 1973; Lorenz, 2002) further study the *so + (adj/adv) + that + a declarative clause* as a significant intensification pattern that has both prototypical syntactic and semantic function as overstatement.

Claridge (2010) find out that in the *so...that* pattern, the content clauses always express the resultant meaning of the prompts and that the clauses itself creates impossible worlds. Such discoveries motivate us to comprehensively uncover the sensical (commonsense or counterfactuality) relationships behind hyperboles in Section 3.2.

8.2 Hyperbole Detection

Troiano et al. (2018) and Kong et al. (2020) explore statistical and neural based approaches to automatic hyperbole detection in English (HYPO-en) and Chinese (HYPO-cn) corpora. Troiano et al. (2018) introduce hand-crafted features while Kong et al. (2020) achieve better performance by jointly training with such hand-crafted features and a directional skipgram. We also train a hyperbole identifier as part of the generation model. However, for our classifier, we finetune the BERT model.

8.3 Figurative Generation

Recent years have witnessed increased interest in creative and figurative language generation. Yu and Wan (2019) generate metaphor unsupervisedly by extracting the metaphorically-used verbs; Chakrabarty et al. (2021) propose a metaphor generation method with symbolism and discriminative decoding; Stowe et al. (2021) study diverse metaphor generation using conceptual mapping. Given a pair of homophones, Yu et al. (2018) train a conditional neural language model with an decoding algorithm for pun generation; He et al. (2019) tackle the same task with a local-global surprisal principle and a retrieve-and-edit pipeline; Luo et al. (2019) on the other hand propose an adversarial pun generative network.

Generating hyperboles or exaggerations is a new task. To the best of our knowledge, we are the first to work on hyperbole generation. The closest work is that of Chakrabarty et al. (2020b), who propose an end-to-end approach for simile generation that also utilizes commonsense knowledge predicted by COMeT (Bosselut et al., 2019). However, they only utilize the `PROPERTY` relation to replace certain parts of literal sentences. We leverage a more complex set of commonsense knowledge during the generation time, and target at a different trope of figurative language.

9 Conclusion and Future Work

We are the first to tackle the novel task of hyperbole generation at the *clause or sentence level* . We start with the representative *so...that* pattern, partition it into three components and analyze the logical relationships among them. Our proposed model **HypoGen** first generates commonsense and counterfactual predictions, and then selects top-ranking candidates as hyperboles. Our experimental results show that **HypoGen**$_{Spec}$ is able to generate hyperboles with high success rate and high semantic intensity, funniness, and creativity scores.

In addition, we propose **HypoPara** as a diversity-oriented generation approach. Follow-up works on hyperbole generation without relying on any patterns can use **HypoPara** as a baseline. Both our **HypoGen** and **HypoPara** can be applied to downstream applications such as dialog systems and storytelling, to improve their interestingness and engagement.

Acknowledgement

We thank the anonymous reviewers and Emily Sheng, Yu (Hope) Hou, and Nuan Wen for their useful feedback. We also appreciate the kind efforts of the undergraduate students from PlusLab to annotate the relations. This work is supported by the Machine Common Sense (MCS) program under Cooperative Agreement N66001-19-2-4032 with the US Defense Advanced Research Projects Agency (DARPA). The views and the conclusions of this paper are those of the authors and do not reflect the official policy or position of DARPA.

Ethics Considerations

We understand and respect user privacy. The HYPO-Red dataset is collected from Reddit totally anonymously, and does not reveal any details about the users' personal information, including name, racial or ethnic origin, religious or philosophical affiliation or beliefs, sexual orientation, etc.

Our proposed methods are based on the pretrained language model. It is known that pretrained language models could capture the bias reflected in the training data (Sheng et al., 2019; Wallace et al., 2019). Considering the nature of exaggeration or overstatement, the context and sentiment of the literal input prompt also affect the our generated hyperboles. Therefore, our models may potentially generate offensive content for certain groups or individuals. We suggest to carefully examine the potential biases before deploying the models to real-world applications.

References

Ulf Bäcklund. 1973. *The collocation of adverbs of degree in English*. Acta Universitatis Upsaliensis.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*.

Tuhin Chakrabarty, Debanjan Ghosh, Smaranda Muresan, and Nanyun Peng. 2020a. R3: Reverse, retrieve, and rank for sarcasm generation with commonsense knowledge. In *the 2020 Annual Conference of the Association for Computational Linguistics (ACL)*.

Tuhin Chakrabarty, Smaranda Muresan, and Nanyun Peng. 2020b. 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)*.

Tuhin Chakrabarty, Xurui Zhang, Smaranda Muresan, and Nanyun Peng. 2021. Mermaid: Metaphor generation with symbolism and discriminative decoding. In *The 2021 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

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. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

He He, Nanyun Peng, and Percy Liang. 2019. Pun generation with surprise. *arXiv preprint arXiv:1904.06828*.

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.

R.J. Kreuz and M.S. MacNealy. 1996. *Figurative language occurrence and co-occurrence in contemporary literature*. Empirical Approaches to Literature and Aesthetics, Norwood, NJ.

Florian Kunneman, Christine Liebrecht, Margot Van Mulken, and Antal Van den Bosch. 2015. Signaling sarcasm: From hyperbole to hashtag. *Information Processing & Management*, 51(4):500–509.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Gunter Lorenz. 2002. Really worthwhile or not really significant? a corpus-based approach to the delexicalization. *New reflections on grammaticalization*, 49:143.

Fuli Luo, Shunyao Li, Pengcheng Yang, Baobao Chang, Zhifang Sui, Xu Sun, et al. 2019. Pun-gan: Generative adversarial network for pun generation. *arXiv preprint arXiv:1910.10950*.

Michael McCarthy and Ronald Carter. 2004. "there's millions of them": hyperbole in everyday conversation. *Journal of pragmatics*, 36(2):149–184.

Laura Cano Mora. 2009. All or nothing: A semantic analysis of hyperbole. *Revista de Lingüística y Lenguas Aplicadas*, 4(1):25–35.

Neal R Norrick. 2012. On the semantics of overstatement. In *Sprache erkennen und verstehen*, pages 168–176. Max Niemeyer Verlag.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. *arXiv preprint arXiv:1909.01326*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Kevin Stowe, Tuhin Chakrabarty, Nanyun Peng, Smaranda Muresan, and Iryna Gurevych. 2021. Metaphor generation with conceptual mappings. In *Proceedings of the Conference of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*.

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.

Jiao Sun, Xuezhe Ma, and Nanyun Peng. 2021. Aesop: Paraphrase generation with adaptive syntactic control. In *The 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. *arXiv preprint arXiv:1908.07125*.

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.

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

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. *arXiv preprint arXiv:1903.00138*.