A Survey on Automatic Generation of Figurative Language: From Rule-based Systems to Large Language Models

HUIYUAN LAI, University of Groningen, Groningen, Netherlands
MALVINA NISSIM, University of Groningen, Groningen, Netherlands

Figurative language generation (FLG) is the task of reformulating a given text to include a desired figure of speech, such as a hyperbole, a simile, and several others, while still being faithful to the original context. This is a fundamental, yet challenging task in Natural Language Processing (NLP), which has recently received increased attention due to the promising performance brought by pre-trained language models. Our survey provides a systematic overview of the development of FLG, mostly in English, starting with the description of some common figures of speech, their corresponding generation tasks, and datasets. We then focus on various modelling approaches and assessment strategies, leading us to discussing some challenges in this field, and suggesting some potential directions for future research. To the best of our knowledge, this is the first survey that summarizes the progress of FLG including the most recent development in NLP. We also organize corresponding resources, e.g., article lists and datasets, and make them accessible in an open repository. We hope this survey can help researchers in NLP and related fields to easily track the academic frontier, providing them with a landscape and a roadmap of this area.

CCS Concepts: • **Computing methodologies** → **Natural language processing**;

Additional Key Words and Phrases: Figurative language, language generation, systematic review

ACM Reference Format:
Huiyuan Lai and Malvina Nissim. 2024. A Survey on Automatic Generation of Figurative Language: From Rule-based Systems to Large Language Models. *ACM Comput. Surv.* 56, 10, Article 244 (May 2024), 34 pages. https://doi.org/10.1145/3654795

1 INTRODUCTION

Figurative language is a cover expression in language studies that includes a variety of figures of speech, such as hyperbole and metaphor, each of which can be used to accomplish a constellation of communicative goals [107]. Figurative expressions can be sentences or even single words, they can make your writing or speech more interesting and captivating, or help convey abstract concepts otherwise difficult to visualize. For instance, in the sentence pair "*He is very happy*" and "*He is floating on cloud nine*," the former simply expresses a state of happiness, while the latter paints a vivid picture, helping people to better understand the intensity of the emotion being conveyed. Understanding and generation of figurative language are particularly challenging due to its implicit linguistic nature and are considered a "*bottleneck*" for automatic language processing [18, 114].

Authors' address: H. Lai and M. Nissim, Center for Language and Cognition, University of Groningen, Harmonie Complex, Groningen 9712 EK, Netherlands; e-mails: h.lai@rug.nl, m.nissim@rug.nl.

ACM 0360-0300/2024/05-ART244
https://doi.org/10.1145/3654795

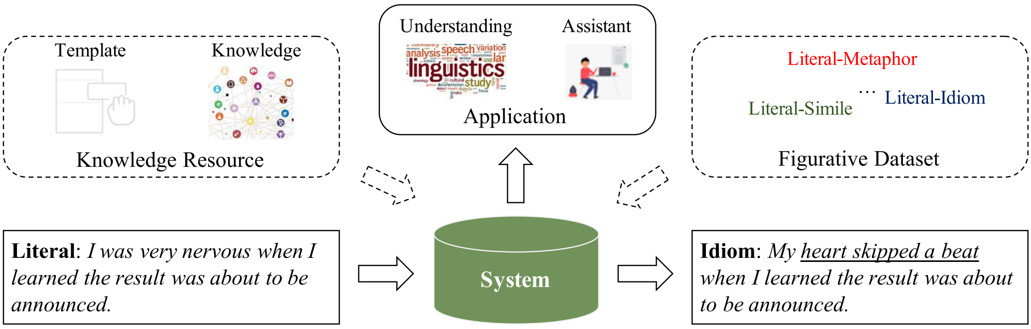

Template
Knowledge
Knowledge Resource
Understanding
Assistant
linguistics
Application
Literal-Metaphor
Literal-Simile
...
Literal-Idiom
Figurative Dataset
Literal: I was very nervous when I learned the result was about to be announced.
System
Idiom: My heart skipped a beat when I learned the result was about to be announced.

Fig. 1. An illustration for figurative language generation from literal text, along with the required data resources and the corresponding applications.

Natural Language Generation (**NLG**) is a fundamental yet challenging branch in **Natural Language Processing** (**NLP**) [36], which aims at automatically generating high-quality, coherent, and understandable human language from various forms of data, such as text [120], image [5], structured data [9, 25]. **Figurative language generation** (**FLG**), also known as creative text generation, is a text-to-text generation task whose goal is to reformulate a given text in a different one containing the desired figure of speech, while still being faithful to the original context (see Figure 1, where the literal snippet "*I was very nervous when I learned the result was about to be announced*" gets rewritten into a sentence which contains the idiom "*My heart skipped a beat when I learned the result was about to be announced.*") This usually requires adding some additional information associated with the original context to trigger and realise the desired figure of speech while at the same time preserving a large portion of the input sentence [19, 105, 146].

Like with other NLG tasks, especially before the advent of **large language models** (**LLMs**, e.g., GPT-3 [16]), researchers have typically relied on knowledge-based approaches for FLG. These kinds of approaches involve a complex process, such as using prior linguistic knowledge to design rules that can capture textual patterns or construct relevant knowledge resources. However, the obtained results are often unsatisfactory, for example lacking the flexibility necessary and linguistic subtlety to instantiate figurative language [115, 140]. With the development of deep learning [41], neural networks achieve impressive performances on various NLP tasks and have become mainstream methods for NLG tasks [67, 89], especially the sequence-to-sequence (seq2seq) framework [120] in the context of rewriting. However, the majority of NLP research has concentrated on literal language, despite the ubiquity of figurative language in human language [18]. Consequently, the advancements in FLG may not have been as substantial as those in other text-to-text generation tasks, such as machine translation [55], question answering [131], and text summarization [24]. The main reason is that most of the earlier works, including the workshop on figurative language processing [11, 39, 54], mainly focused on figurative language detection rather than generation. Moreover, the NLP community lacks standard benchmark datasets for FLG, which is a crucial point to advance a research field.

Recent years have witnessed the emergence and rapid advancement of **pre-trained language models** (**PLMs**), in particular the commonly used Transformers [128] based models [16, 27, 65, 99, 101], which have become the most popular method in NLP and can achieve state-of-the-art performance in virtually all tasks. In this paradigm, neural models are pre-trained on large-scale unlabeled text collections in a self-supervised fashion to learn fundamental language representations. Then, the model can be fine-tuned for downstream tasks with task-specific training objectives, thus avoiding the need to train a new model from scratch. Recently, researchers apply

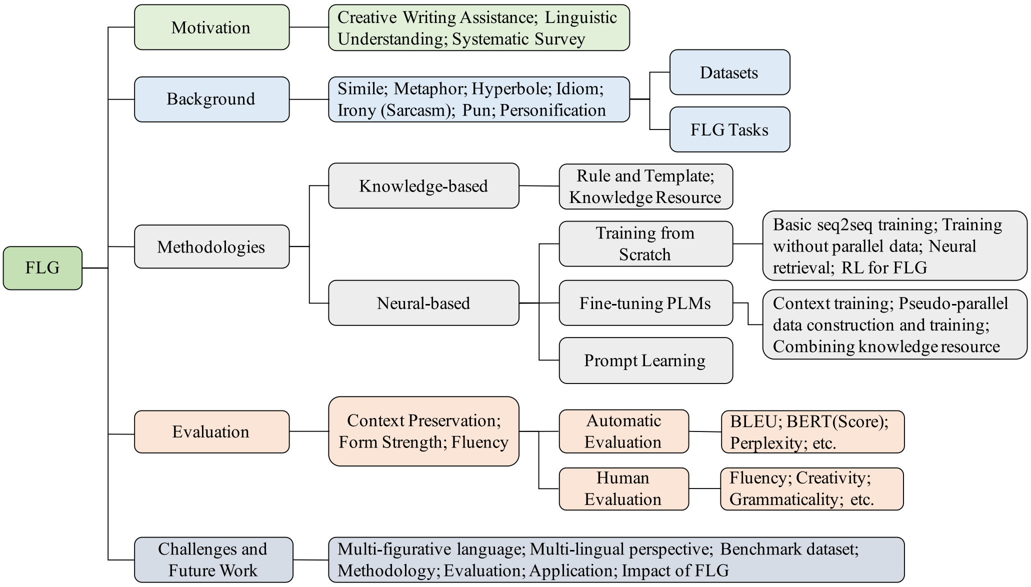

FLG
Motivation
Creative Writing Assistance; Linguistic Understanding; Systematic Survey
Background
Simile; Metaphor; Hyperbole; Idiom; Irony (Sarcasm); Pun; Personification
Datasets
FLG Tasks
Methodologies
Knowledge-based
Rule and Template; Knowledge Resource
Neural-based
Training from Scratch
Basic seq2seq training; Training without parallel data; Neural retrieval; RL for FLG
Fine-tuning PLMs
Context training; Pseudo-parallel data construction and training; Combining knowledge resource
Prompt Learning
Evaluation
Context Preservation; Form Strength; Fluency
Automatic Evaluation
BLEU; BERT(Score); Perplexity; etc.
Human Evaluation
Fluency; Creativity; Grammaticality; etc.
Challenges and Future Work
Multi-figurative language; Multi-lingual perspective; Benchmark dataset; Methodology; Evaluation; Application; Impact of FLG

Fig. 2. Overview of the survey.

PLMs to FLG including the generation of metaphor [1, 117], hyperbole [123, 145], simile [21, 143], irony [19, 148], idiom [146], pun [77, 141], analogy [12], and personification [76] (see Section 2.2 for more details on these figures of speech). Given the rapid development of FLG, a comprehensive literature review can help more NLP researchers and practitioners to participate and easily track the academic frontier in this area, especially making it more accessible for interested beginners to enter this field. Additionally, achieving the goal of enabling machines to write in different figures of speech has a wide range of applications, which are mainly divided into two categories: aiding in various downstream NLP tasks [31] and supporting the development of application products such as educational systems [21].

As far as we know, this is the first survey that presents a comprehensive review of FLG. It first introduces the background of different figures of speech, and surveys related generation tasks as well as their corresponding benchmark datasets. We find that existing works mainly focus on modelling single figurative forms, i.e., rewriting a literal sentence into one containing a specific figure of speech [21, 72, 95, 115, 124]. Then, we review the modelling approaches, from traditional to state-of-the-art, and divide them into two categories: knowledge-based and neural-based approaches. Regarding knowledge-based approaches, they usually require linguistic knowledge to design rules and templates to fit figurative patterns [1, 121], or to construct relevant resources for modelling [45, 91]. Neural approaches, instead, require less linguistic knowledge and complex linguistic processing but need large-scale labelled data (e.g., parallel literal-hyperbole pairs) and computational resources for training neural models. We also review the various assessment strategies for FLG, which include both human-based and automatic evaluation, discussing their corresponding pros and cons. Finally, based on our review and the analysis of current trends, we suggest several challenging problems and prospective research directions.

As shown in Figure 2, the main contributions of this survey can be summarized as follows:

- **Systematic Survey.** We offer the first comprehensive review of FLG, which includes background information for common figures of speech, benchmark datasets, modelling approaches, and evaluation methods.

Table 1. Examples of Figurative Language Generation from Literal Texts

Figure of speech	Sentences
Literal	From the day you were born, you have been invincible.
Simile	From the day you were born, you have been like a well-seasoned superhero.
Literal	Life has good and bad moments, often in fast succession.
Metaphor	Life is a roller coaster with lots of ups and downs.
Literal	My new laptop is very thin and light compared to the bulky old one.
Hyperbole	My new laptop looks like a piece of paper compared to the bulky old one.
Literal	We were all anxious as we waited for the surgeon's report.
Idiom	We were all on pins and needles as we waited for the surgeon's report.
Literal	I hate when people don't think the rules apply to them.
Irony (Sarcasm)	I love when people think the rules don't apply to them.
Literal	Everyone should learn to be satisfied with what they have.
Pun	Everyone should learn to be satisfied with the state they are in.
Literal	The birds chirping in spring.
Personification	The birds welcome the spring by singing melodious tunes.

- **Available Resources.** Based on the literature review, we collect and make accessible all available resources described in this survey (including corpora and paper lists), so as to foster new work in this field.[1]
- **Future Directions.** We discuss current progress and the corresponding limitations, thereby offering an outline of possible future research directions in FLG. We also suggest some potential solutions for a number of open issues.

The rest of the survey is organized as follows: In Section 2, we introduce background information for several figures of speech, along with some related FLG tasks and datasets. In Section 3 and 4, we discuss knowledge-based and neural-based approaches. In Section 5, we describe evaluation strategies, including both automatic and human-based methods. In Section 6, we bring forward the main challenges and limitations of FLG, and suggest improvements for future work in this area. We conclude the article with a short reflection in Section 7.

2 BACKGROUND

We first provide an overview of FLG from a conditional distribution perspective, then introduce some background information for several figures of speech commonly studied in NLP, as well as their corresponding datasets in the deep learning era. We also discuss two closely related tasks, namely figurative language detection and text style transfer.

2.1 Figurative Language Generation

Figurative language generation refers to the task of reformulating a given text into one containing the desired figure of speech while still being faithful to the original content. Table 1 shows examples of rewriting the given literal texts in the target figurative form.

For a source sentence $\mathbf{x} = \{x_1, \ldots, x_n\}$ of length n, the goal is to generate a sentence $\mathbf{y} = \{y_1, \ldots, y_m\}$ of length m with a specific figurative form f. Formally, the generative model of the

[1] https://github.com/laihuiyuan/figurative-language-generation

conditional probability of the observable $\mathbf{x}$, given a target $\mathbf{y}$, can be formulated as

$$p_\theta(\hat{\mathbf{y}}|\mathbf{x}, f) = \prod_{t=1}^{m} p_\theta(\hat{y}_i|\hat{y}_{1:i-1}, \mathbf{x}, f). \tag{1}$$

The target form f may vary according to the target figure of speech. To be considered successful, the generated sentence $\hat{\mathbf{y}}$ must satisfy three criteria: (i) it must contain the target figurative form f; (ii) it must preserve the original context; and (iii) it must exhibit appropriate natural language characteristics such as fluency, readability, and coherence. It is worth noting that there are two possible ways of generation: (i) the generated sentence only contains the target figurative form while the original form is removed; or (ii) the generated sentence contains both the source and target figurative forms. For example, the hyperbolic sentence "*I am not happy that he urged me to finish all the hardest tasks in the world*" can be rewritten as "*Glad he urged me to finish all the hardest tasks in the world*," which contains both sarcasm and hyperbole.

2.2 Figures of Speech and Datasets

Table 2 shows the related FLG tasks and their corresponding datasets. The FLG tasks mainly focus on rewriting literal sentences into sentences with specific figures of speech, plus a few other tasks, such as generating a follow-up sentence that is coherent with the narrative and consistent with the figurative expression. As far as language is concerned, most datasets are in English, with a few in Chinese and German. Overall, each figure of speech contains at least one related task and one dataset, which are detailed below.

Simile. It is a figure of speech that compares two different things by saying that one thing is like another one, so it often uses comparison expressions such as *like*, *as*, and *than*. For instance, the sentence "*From the day you were born, you have been invincible*" could be rephrased as "*From the day you were born, you have been like a well-seasoned superhero.*" Here, using the simile "*like a well-seasoned superhero*" can make the expressions and descriptions more emphatic (it is often used in literature and poetry to spark the reader's imagination [94]).

To create a simile dataset, Chakrabarty et al. [21] collected sentences from the web containing the phrase *like a*. Then, the authors employed the generative model of the knowledge graph COMET [13] transforming English similes into their corresponding literal sentences, thereby automatically creating literal-simile pairs for supervised training. Zhang et al. [143] created a large-scale simile dataset in Chinese, including similes and corresponding contexts containing similes, from online free-access fiction. Chakrabarty et al. [18] released a simile-related dataset, which aims at generating a plausible next sentence that is coherent with the context and consistent with the meaning of the simile that follows the given narratives. More recently, Bhavya et al. [12] proposed the task of generating (i) a source concept analogous to a given target concept, and (ii) an explanation of the similarity between the target and the source concept. Given that this dataset contains a large number of "*like*" expressions, we consider it in the simile category. Similarly, Chakrabarty et al. [22] introduced FLUTE, a dataset containing samples of literal, and figurative (simile, metaphor, and sarcasm) sentences along with their natural language explanation.

Metaphor. This is the most common figure of speech [58], which refers to one concept, usually more abstract, by means of another one, usually more concrete. Unlike similes that explicitly say that one thing is like another, a metaphor conveys that one thing is the same as another at some level. The literal sentence "*Life has good and bad moments, often in fast succession*," for instance, can be reformulated with a metaphor as "*Life is a roller coaster.*" Here, the verb *be* (i.e., *is*) is used instead of *like*, and offers a vivid and concrete way (i.e., a roller coaster) to visualise life. Metaphor is not only a form used to add a creative flavour to the text, but can also be utilized to achieve textual goals that may be difficult to obtain through literal expressions [50].

Table 2. List of Common Figures of Speech and their Related Datasets

Figure of speech	Task	Dataset	Train	Valid	Test	Lang	Para
Simile	Literal↔Simile	Chakrabarty et al. [21]	82,687	5,145	150	en	✓
	Simile↔Context with simile	Zhang et al. [143]	5.4M	2,500	2,500	zh	✓
	Narrative+Simile→Text	Chakrabarty et al. [18]	3,100	376	1,520	en	✓
	Concept→Analogy + Explanation	InstructGPT [12]	–	–	148	en	✓
	Simile + Literal→Explanation	FLUTE [22]	1500	–	–	en	✓
Metaphor	Literal↔Metaphor	Stowe et al. [115]	260k	15,833	250	en	✓
		Chakrabarty et al. [23]	90k	3,498	150	en	✓
		Stowe et al. [117]	248k	–	150	en	✓
		Mohammad et al. [85]	–	–	171	en	✓
		CMC [69]	3,554/2,703	–	–	en	×
	Metaphor + Literal→Explanation	FLUTE [22]	1500	–	–	en	✓
Hyperbole	Literal↔Hyperbole	HYPO [124]	709	–	–	en	✓
		HYPO-cn [56]	2,082/2,680	–	–	zh	×
		HYPO-red [123]	2,163/1,167	–	–	en	×
		HYPO-XL [145]	-/17,862	–	–	en	×
Idiom	Idiom→Literal	Liu and Hwa [72]	88	–	84	en	✓
	Idiom (en)↔Literal (de)	Fadaee et al. [31]	1,998	–	1,500	en/de	✓
	Idiom (de)↔Literal (en)		1,848	–	1,500	de/en	✓
	Literal↔Idiom	PIE [146]	3,784	876	876	en	✓
	Narrative+Idiom→Text	Chakrabarty et al. [18]	3,204	355	1,542	en	✓
Irony (Sarcasm)	Literal↔Irony(Sarcasm)	Peled and Reichart [95]	2,400	300	300	en	✓
		Mishra et al. [83]	–	–	203	en	✓
		Zhu et al. [148]	112k/262k	–	–	en	×
		Ghosh et al. [40]	4,762	–	–	en	✓
	Sarcasm + Literal→Explanation	FLUTE [22]	3356	–	–	en	✓
	Image→Sarcastic Description	SentiCap [79]	–	–	503	en	✓
Pun	Word Senses→Pun	SemEvaltask7 [82]	1274	–	–	en	✓
	Context→Pun	Sun et al. [119]	2,753	–	–	en	✓
Personification	Topic→Personification	Liu et al. [76]	67,441	3,747	3,747	zh	✓

Notes: (i) Lang = Language, Para = Parallel training data; (ii) all datasets reported here are mainly used to train neural network based models.

Veale [129] states that computational metaphor generation can be used for many applications, such as entertainment, education, and even pure whimsy. Of all the figures of speech, metaphor receives the most attention from researchers. However, existing metaphor datasets for automatic generation are almost all in English. For instanceMohammad et al. [85] released a metaphor evaluation dataset in English. They first extracted metaphorical and literal verb senses from WordNet [81], then created datasets of pairs of their usages by asking crowd workers to annotate the usage of verb occurrences as metaphorical or literal. Stowe et al. [117] and Chakrabarty et al. [23] built two large-scale literal-metaphor datasets by exploiting the Gutenberg Poetry corpus [49]. Specifically, they first trained a word-level metaphor detection classifier to identify and mask the metaphoric verbs in the sentences, and leveraged the pre-trained language model, which uses the unmasked context to predict the masked word. Similarly, Stowe et al. [115] used a mask-then-fill method to create a literal-metaphor dataset. They built a literal vocabulary and a metaphoric vocabulary using lexical resources (MetaNet [28] and FrameNet [6]), which were then used to mask the words of given sentences and select the best-fit words in the filling process.

Hyperbole. It is a figurative form in which an expression is deliberately exaggerated and exceeds the credible limits of fact in the given context. It is the second most common form, after metaphor [58, 87], and is generally used for two main expressive purposes: (i) emphasizing your argument, e.g., the hyperbole "*My son takes years to finish his homework*" where the time taken to finish homework is not years: it is just a way to make the expression more impactful; and (ii) comparing or highlighting the differences between two things, e.g., the hyperbole "*My new laptop looks like a piece of paper compared to the bulky old one*," where the aim is at highlighting the different weigh between the two laptops by comparing one with a piece of paper in an exaggerated way.

There are four datasets related to the hyperbole generation task. HYPO [124] is an English parallel dataset, which contains 709 hyperbolic sentences, each with a manually rewritten non-hyperbolic counterpart. HYPO-red [123] and HYPO-XL [145] are both English non-parallel datasets, the former is created by asking annotators to label the exaggerated attributes of sentences, while the latter is created in a semi-supervised way that uses a binary classifier to predict possible hyperbolic sentences. HYPO-cn [56] is a manually created Chinese parallel dataset.

Idiom. An idiom is a figure of speech in which a group of words has established meaning over a long period of usage that in most cases cannot be deduced directly from the individual words in the expression, i.e., it is not computational [87]. For example, the sentence "*We were all on pins and needles as we waited for the surgeon's report*" indicates people were all tense rather than literally sitting on pins and needles. Research work found idioms are usually problematic for (second) language learners [72, 136], because they are semantically opaque and very much language- and culture-dependent. Similarly, idiom understanding and generation is also a challenging problem in NLP tasks such as sentiment analysis and machine translation [57, 132, 136]. More concretely, Salton et al. [110] found that a machine translation system may only perform half as well on texts containing idioms compared to those without idiomatic expressions. Volk and Weber [132] even argued that translating idioms is one of the most difficult tasks for both human and automatic translations. Therefore, systems for idiom translation and generation could be used to assist in understanding and creative writing, especially for beginners and second language learners.

For the task of idiom generation, Fadaee et al. [31] constructed a dataset for idiom translation based on WMT's German-English pairs from 2008 to 2016. Specifically, they used `dict.cc`, an online dictionary containing idiomatic and literal phrases, to select sentence pairs from the dataset whose source sentences contain idioms. Liu and Hwa [72] proposed a task of replacing idioms with literal English. To build a dataset, they randomly selected Tweets that contain idioms and usage examples from `TheFreeDictionary.com`. Finally, they presented these sentences along with each idiom's definition to native speakers of English and asked them to manually shorten the definition. Zhou et al. [146] released a parallel dataset containing their manually created textual counterparts for idiomatic texts from the existing corpus MAG-PIE [42]. Chakrabarty et al. [18] released a manually created parallel English dataset for generating a target sentence that is coherent with the context and consistent with the meaning of the idiom. Recently, Stowe et al. [118] introduced a new natural language inference task for figurative language and created an English dataset consisting of paired sentences spanning idioms and metaphors, which could be a valuable resource for FLG.

Irony (Sarcasm). This is a figure of speech that can make the literal sentiment of the text different from the implied intent, often with an element of hostility, irritation, or just fun. Eric Partridge writes in Usage and Abusage: "*Irony consists in stating the contrary of what is meant.*" [93] Therefore, it usually takes longer for people to understand irony than to understand literal expressions [111]. For instance, the sentence "*I hate when people don't think the rules apply to them*" can also be expressed in an ironic way "*I love when people think the rules don't apply to them.*" It is worth briefly unpacking the relationship between irony and sarcasm, often considered a "type" of irony [14, 59, 64]. While irony is not necessarily negative (though the intended sentiment is mainly this), nor intentional, sarcasm is generally an intentional and negative form of verbal communication alternative to direct criticism.

In terms of datasets used in generation, Peled and Reichart [95] built a parallel dataset consisting of 3,000 sarcastic tweets, each augmented with five corresponding non-sarcastic ones rewritten by crowd workers. Mishra et al. [83] created a dataset for evaluation that contains 203 sentence pairs in which sarcastic utterances were manually translated into literal versions by linguists. Ghosh et al. [40] used crowdsourcing to create a parallel dataset containing explicit interpretations of verbal irony. Zhu et al. [148] trained an irony classifier to partition the sentences into ironic and non-ironic, automatically creating a large-scale non-parallel tweet dataset. Ruan et al. [109] recently introduced a multimodal sarcasm generation task, i.e., generating a sarcastic description for a given image, using the testing subset of 503 images in the dataset SentiCap [79].

Pun. A pun, also known as paronomasia, is a form of wordplay in which a word suggests multiple meanings by exploiting polysemy, homonymy, or phonological similarity to others, with the purposed aim of yielding a humorous expression. The sentence "*Everyone should learn to be satisfied with the state they are in*," for example, "*state*" here can be an organized political community forming part of a country, while it can also be the mode or condition of someone, triggering an ambiguity that in specific contexts could be perceived as appropriately amusing, or clever.

Miller et al. [82] released a SemEval-2017 dataset for the detection of English Puns. This contains human-written puns annotated with pun words and alternative words, which has been used as a benchmark in the pun generation task to test the models' performance [77, 139]. However, most pun pairs only occur once in the dataset, while one given context could have been compatible with many other pun pairs. Based on the SemEval-2017 dataset, Sun et al. [119] constructed a new dataset containing 4,551 tuples of context keywords and an associated pun pair, each labelled with whether they are compatible for composing a pun or not, along with 2,753 human-written puns for the compatible pairs. For example, the tuple *context: construction workers; pun pair: stair/stare* is compatible, allowing for the creation of the pun "*Two construction workers had a staring contest.*"

Personification. It is a figure of speech similar to the anthropomorphic metaphor in literature or art, which involves attributing human characteristics to non-human entities. This, therefore, allows people to create life and motion within inanimate objects, animals, and even abstractions by assigning them recognizable human behaviours and emotions. In other words, personification can provide readers with vivid human-like characteristics, ultimately making the expression more concrete and empathic. For example, the sentence "*The birds welcomed the spring by singing melodious tunes*" gives the bird human behaviour of welcoming spring.

For the task of personification generation, Liu et al. [76] created a parallel personification dataset, which is used for generating modern Chinese poetry for the given topics, while controlling for the use of metaphor and personification.

2.3 Related Tasks

Figurative language generation, as a subfield of figurative language processing, involves the creation or transformation of figures of speech when text is rewritten, where the generated text should contain specific figurative forms. A good number of NLP tasks could be conceived as related to FLG, namely anything which involves automatic language generation and non-literal meaning, thus including machine translation, summarisation, the automatic treatment of creativity, and so on. However, for a closer peek into the phenomenon at hand, here we mainly discuss the background of two tasks that we deem as most related to and relevant for FLG: one is figurative language detection which is related to figurative language itself, and the other is style transfer which is related to stylistic text rewriting.

Figurative Language Detection. This task typically involves two levels of detection, the word level, which identifies the exact words within a sentence that trigger the figurative attribute, and the sentence-level detection, which determines whether a sentence is literal or non-literal. It is important to note that word-level detection is a crucial component in retrieval-based FLG models, as these methods typically require first identifying trigger words in sentences, followed by other operations such as replacement and generation. Over the past two decades, numerous approaches, mostly based on machine learning algorithms, have been proposed to address the problem of figurative language detection. With traditional machine learning methods, researchers must define and identify linguistic features, which are then fed into the model to learn task patterns, such as the recognition of metonymy [86], hyperbole [124], metaphor [10, 78, 125], idiom [68], irony [108]. Neural networks, instead, can achieve impressive results in figurative language detection without the need for feature engineering, such as using convolutional neural network [29, 102], LSTM [33, 138], and the pre-training model BERT [26, 127], and mT5 [63].

Text Style Transfer. Text style transfer is the task of transforming a text of one style into another while preserving the original content. For example, given the informal sentence "*different from what I've seen*," we can turn it into the formal counterpart "*It is different from what I have seen*" for the task defined as formality transfer [104]. Text style transfer is very similar to FLG as they both aim at achieving the generation of text with desired attributes, the former being a specific style while the latter a figurative form. However, there are also several differences between the two tasks in terms of the types of text changes involved. For instance, text style transfer may require modifying multiple parts of a sentence simultaneously, such as capitalization at the beginning, punctuation at the end, and some phrasing in the middle. In contrast, FLG often involves rewriting only certain expressions to trigger the figure of speech, while retaining other large parts of the sentence unchanged [146]. Most importantly, in text style transfer the original style must be completely changed, whereas in FLG the original figurative form could be still present in the generated sentence (see the example in Section 2.1).

3 KNOWLEDGE-BASED APPROACHES

In earlier works, as with other NLG tasks, FLG research mainly focused on knowledge-based approaches. Generally, NLP researchers need to master linguistic knowledge to design rules and templates to fit figurative language patterns, or to construct the corresponding knowledge resources. We roughly divide knowledge-based approaches into two sub-categories: (i) rule (template) and (ii) knowledge resources. Their corresponding classic works, advantages (pros) and disadvantages (cons) are presented in Table 3 (first block).

3.1 Rule and Template

Rule- and template-based generation is a simple yet efficient method consisting of a set of predefined modules that represent cues and indicators of figurative forms. These types of systems often do not require a training process and can generate the desired figurative language quickly and efficiently once the templates are created.

A representative method is the template "*A (e.g., vehicle) is like B (e.g., topic)*" used to acquire relations between A and B which can be used for metaphor generation [1, 121]. The computational systems are developed to generate simple metaphors, which are based on the probabilistic relationships between words in the textual data. Similarly, Veale [129] used XYZ comparisons with the template form "*X is the Y of Z*" to generate creative and metaphoric tweets. In this template, Y refers to a proper-named individual that represents an entire class of people in a figurative sense.

Table 3. A Summary of the Two Main Approaches to Figurative Language Generation

Type	Subcategory	Pros	Cons	References
Knowledge-based	Rule and template	- Intuitive and simple - Tailored to a specific form	- Poor flexibility and diversity	[1, 51, 121, 129]
	Knowledge resource	- Exploiting knowledge resource - High interpretability	- Prior linguistic knowledge - Construct desired resources	[47, 96, 97, 130] [38, 72, 113, 126] [43, 45, 91, 117]
Neural-based	Training from scratch	- Straightforward - Combine retrieval approaches	- Large-scale training data - Large computational resources	[31, 76, 95] [44, 69, 139, 140] [77, 141, 146, 148] [83]
	Fine-Tuning PLMs	- Straightforward - Pre-trained knowledge - State-of-the-art results	- Large computational resources	[21, 143, 145, 147] [23, 115, 117] [18, 19, 84, 123] [61]
	Prompt Learning	- Straightforward - A few/no labelled samples	- Prompt engineering - Large computational resources	[12, 18, 84, 106]

We list their subcategories, along with corresponding pros and cons, and some classic references.

For example, the sentence "*the potato is the Tom Hanks of the vegetable world*," where Hanks (Y) is the representative of the versatile actor. The authors collected a vast amount of tuples of the form XYZ from the Internet, using the capitalization of the Y field as a cue to find those that pivot around people. Instead of using a single template, Joshi et al. [51] designed a system consisting of eight rule-based modules, each targeting a certain type of sarcastic expression, to generate sarcasm responses for given user inputs.

In summary, rule- and template-based FLG can be customized to specific figures of speech, making the whole process pretty transparent. However, this method is extremely inflexible, as humans must manually design rules and templates for each figure of speech separately. Additionally, it is primarily geared toward the most common figurative patterns and thus lacks diversity, and certainly does not take full advantage of empirical knowledge of the language.

3.2 Knowledge Resources

This type of approach often involves the operations of search, identification, replacement, and mapping done by utilizing existing lexical resources or self-built knowledge resources.

WordNet. One example of the widely used lexical resource for FLG is WordNet [81]. Based on two specific templates "*X is the Y of Z*" and "*X is as Y as Z*", Pereira et al. [96] employed WordNet and structure mapping algorithms to enhance texts with various figures of speech. With WordNet, the lexical realization process is enriched by lexical selection, where a decision is made between lexical alternatives representing the same content. Petrović and Matthews [97] proposed an unsupervised system to generate "*I like my X like I like my Y, Z jokes*". The X, Y, and Z here are variables to be filled in using WordNet and Google n-gram data [2]. Veale and Hao [130] presented a WordNet-based framework, which allows for a concise and adaptable representation of concepts for both metaphor interpretation and generation. Hong and Ong [47] developed a system that utilizes WordNet and ConceptNet [74] to automatically extract word relationships, such as synonyms, hypernyms, sounds-like, and semantic relationships in puns, and stored them in templates. Using these templates and linguistic resources, the system can generate puns starting with a keyword input from the user. Valitutti et al. [126] proposed a lexical replacement approach for generating humour that incorporates a simple form of punning: they created word substitution blocks using WordNet to introduce incongruity and taboo words, thereby attempting to elicit a sense of humour. Shutova [113] proposed an approach for metaphor interpretation, which generates literal paraphrases for given metaphorical expressions. Concretely, for a metaphorical expression, the system (i) generates a list of possible paraphrases that might appear in the same context based on a large corpus; (ii) ranks these generated paraphrases according to their likelihood derived from the corpus; (iii) discriminates between literal and figurative paraphrases by detecting selectional preference violations; and (iv) disambiguates the sense of the literal paraphrases using WordNet's inventory of senses.

Other Resources. Researchers also employed other linguistic resources to develop the FLG system. Liu and Hwa [72] proposed a retrieval-then-replacement approach to translate idioms into literal paraphrases. They first used `TheFreeDictionary.com` to retrieve the related replacement phrase for the idiom, and then replace the idiom with the retrieved phrase through appropriate grammatical and referential transformations. Gero and Chilton [38] used the open-source knowledge graph ConceptNet and a modified Word Mover's Distance algorithm to develop a metaphorical writing collaborative system. This system can provide users with a large, rank list of suggested metaphorical connections based on their input, and generate coherent content with the given context, thereby helping users to produce more diverse texts. Recently, Stowe et al. [117] proposed an unsupervised model for metaphor generation, which is based on frame embeddings [6]. The authors employed FrameNet to learn the joint embedded representations for domains and lexical entities, which are used to represent conceptual domain mappings, and can be applied to generate a word in the target domain for the given input words in the source domain, which is a typical mechanism for metaphor creation.

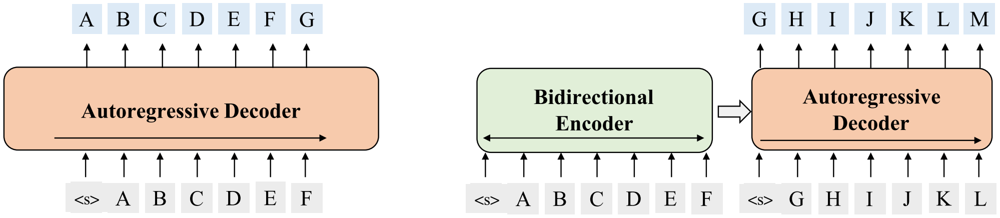
Creating Resources. NLP researchers have also created dedicated resources for FLG, in addition to leveraging existing resources. For instance, Hervás et al. [45] built the mappings between the concepts of source and target domains in metaphors, which are used to generate a set of possible metaphorical references for each concept. They then assessed the clarity and suitability of these metaphorical references to filter the inappropriate ones. The remaining metaphors are used to refer to the concept in a given context only if there is no loss of meaning or unnecessary ambiguity. Since some common sense knowledge can be approximated by parsing a sentence and abstracting from its syntactic structure, Ovchinnikova et al. [91] adopted this approach to design proposition databases, which are used for generating conceptual metaphors and automatically finding the corresponding linguistic expressions in corpora. To reduce the manual workload, Harmon [43] developed a web-driven approach to form a preliminary knowledge base of nouns and their characteristics, and employed it to generate metaphors and similes with various properties like clarity and novelty.

In summary, these kinds of approaches can leverage available language knowledge resources such as WordNet and FrameNet to build FLG systems. The language generation process of such systems is usually interpretable, but they still lack flexibility since new resources, or lexical mappings and constraint conditions, must be designed by experts, usually ad-hoc.

4 NEURAL-BASED APPROACHES

In recent years, neural networks have become the main methods for language generation tasks thanks to the growth of computational power and the availability of large-scale data. As depicted in Figure 3, neural language generation models can be broadly classified into two categories: decoder-only and sequence-to-sequence (seq2seq) architectures. In the following sections, we will provide a brief overview of each.

The GPT family model [16, 98, 99] is representative of decoder-only models, consisting of the decoder component only. These models are mainly used to predict future behaviour based on past observations, i.e., the next generated token is based on what has been seen until then in the sequence. In short, this can be considered as iteratively predicting the probability of the next token,



(a) Decoder-only architecture.

(b) Seq2seq architecture.

Fig. 3. Overview for decoder-only and seq2seq architectures.

which can be formally described as

$$p(x_1, \ldots, x_m) = \prod_{i=1}^{m} p(x_i | x_1, \ldots, x_{i-1}). \tag{2}$$

Where $x_{1:i-1}$ is the sequence of tokens preceding the ith time step, which is used to predict the token for that step. Since tokens are predicted autoregressively, these kinds of models are naturally suitable for text generation tasks.

Unlike decoder-only models, seq2seq models contain both an encoder and a decoder, where the former encodes the source text and the decoder is used for the autoregressive generation. This thus unifies language understanding and generation in a single framework. The seq2seq models are usually based on architectures such as **recurrent neural network** (**RNN**) [120], **convolutional neural network** (**CNN**) [37], and Transformer [128]. Given a source sentence $\mathbf{x} = \{x_1, \ldots, x_m\}$ and a target sentence $\mathbf{y} = \{y_1, \ldots, y_n\}$, the encoder first learns to encode a variable-length sequence of source sentences into a fixed-length vector representation, which can be formulated as

$$(\mathbf{e}_1, \ldots, \mathbf{e}_m) = \text{Encoder}(\mathbf{w}_1, \ldots, \mathbf{w}_m), \tag{3}$$

where $\mathbf{w}_i$ is the fixed-length embedding of token x_i, and $\mathbf{e}_i$ is its corresponding contextualized hidden representation. After that, the decoder is used to decode the vector representation into a variable-length sequence. Specifically, the decoder generates a token at each time step like the decoder-only model, which consumes the previously generated sequence as additional input. This process can be represented as

$$\mathbf{d}_i = \text{Decoder}(\mathbf{e}_{1,\ldots,m}, \hat{\mathbf{w}}_{1,\ldots,i-1}), \tag{4}$$

$$p_\theta(\hat{y}_i | \hat{y}_1, \ldots, \hat{y}_{i-1}) = \text{Logit}(\mathbf{d}_i), \tag{5}$$

where $\hat{\mathbf{w}}_i$ is the fixed-length embedding of generated token $\hat{y}_i$, and Logit($\cdot$) is a nonlinear multi-layered function that predicts the probability of output $\hat{y}_t$. A seq2seq architecture is commonly used in sequence learning and has shown great success in various text-to-text generation tasks [24, 65, 101, 131].

In the field of FLG, almost all neural-based models are based on these two structures. We classify all the models into three categories based on their training methods: (i) training from scratch; (ii) fine-tuning PLMs; and (iii) prompt learning. In the following, we will review the related FLG literature and discuss their advantages and disadvantages in detail.

4.1 Training From Scratch

Training a neural model from scratch involves initializing a new neural network randomly (which may be based on an existing architecture) and then training it for a target task using the corresponding training objective and dataset. In this context, the **long short-term memory** (**LSTM** [46]) network with the attention mechanism [4, 53] is widely used and very efficient for language generation tasks. In the seq2seq framework, the attention mechanism is employed to allow the decoder to use the most relevant parts of the input sequence in a flexible way, by a weighted combination of the output from the encoder, with the most relevant parts being attributed to the highest weights.

Basic seq2seq training. Generally, a seq2seq framework needs to be trained with parallel pairs containing source and target texts. For instance, Peled and Reichart [95] used literal-sarcastic pairs to train an LSTM-based seq2seq model for sarcasm utterance interpretation, which aims at translating sarcastic texts into non-sarcastic. Fadaee et al. [31] trained an attention-based LSTM seq2seq model using parallel data to perform the task of idiom translation between English and German. Liu et al. [76] presented a seq2seq model for metaphor and personification generation based on poetry topics. In order to control the rhetoric modes, they introduced additional strategies to capture various rhetorical patterns and regulate the generation process. Stowe et al. [116] trained a seq2seq model with synthetic parallel data, where the authors employed a metaphor masking framework for metaphor generation. They replaced metaphoric words in the input texts with unique metaphor masks, resulting in parallel training data consisting of the source sentence with masked words and the target sentence with the original words (e.g., "*The war <MET> many people*"→"*The war uprooted many people*"). At inference time, they fed the verb-masked literal texts into the model trained with metaphor masking to generate the target outputs.

Training without parallel data. Supervised learning usually requires large-scale parallel training data, but the creation procedure is very time-consuming and costly, especially for different figures of speech. In light of this, many unsupervised approaches with no need for manually labelled parallel data are commonly used. For instance, Yu et al. [139] proposed a model for generating homographic puns without using any pun training data. Specifically, they first trained a conditional neural language model from a general text corpus, which can generate a sentence containing a given word with specific senses. Then, based on two senses of a target word as inputs, they designed a novel joint beam search algorithm to generate two pun sentences in parallel. The two pun sentences should be the same except for the input words, and are suitable for two specified senses of a homographic word (e.g., For the pun "*Math teachers have lots of problems*," the "*problem*" here can refer to either (i) a source of difficulty; or (ii) a question proposed for consideration or solution.) Yu and Wan [140] presented a framework for verb-oriented metaphor generation, which is trained on an English Wikipedia corpus without manually labelled metaphor data. They designed an unsupervised approach to automatically extract the metaphorically used verbs and their fit words. For example, in "*she devoured (enjoyed) his novels*," the literal sense of the fit word "*enjoyed*" represents the sense of "*devoured*" in this context. Then, they employed a POS-constrained language model to generate a sentence containing the given verb, while considering its fit word in the decoding using a specifically dedicated algorithm. Li et al. [69] presented a GPT-2 based multitask framework for metaphor generation. Specifically, they used a small amount of metaphor-labelled data to train a metaphor identification module on top of the GPT-2 contextualized embedding, which is then used to obtain potential metaphors from a large-scale unlabelled corpus. Based on the identification module that can compute the metaphorical probability, a metaphor weighting mechanism is designed to encourage the generation model to pay more attention to metaphor-related parts of the input, thereby improving the metaphoricity of the generated text.

Neural retrieval. A neural-based retrieval approach has also been explored in FLG, which generally does not require parallel training data. This approach often involves a series of steps, such as retrieval, replacement, and generation. He et al. [44] introduced a local-global surprisal principle in pun generation, where they hypothesized that there is a strong association not only between the pun word and the distant context but also between the corresponding alternative word and the immediate context. For example, the text "*Yesterday I accidentally swallowed some food colouring. The doctor says I'm OK, but I feel like I've dyed (died) a little inside*," the pun word "*dyed*" indicates that the person is coloured inside by food colouring, while the alternative word "*died*" implied another interpretation in context: the person could be dying due to the accident. This contrast can create surprise, which in turn creates a sense of humour. To instantiate the local-global surprisal principle, they introduced an unsupervised approach based on a retrieve-and-edit framework that was developed to replace the alternative word with the pun word, thereby generating puns starting from an unhumorous corpus. However, such direct replacement usually leads to grammatical errors in the generated sentences since the part-of-speech tags of puns and their alternative words are often different. With candidate sentences containing the alternative word at hand, Yu et al. [141] proposed a seq2seq framework with lexical constraints to address this issue. For a given pair of homophones, they first retrieved sentences containing the pun word and the alternative word. Then, they designed a selection algorithm to extract positive constraints between the pun word and the support word in the corresponding sentence (e.g., the pun word "*tuna*" and the support word "*fisherman*" from the sentence "*the fisherman catch tuna with several methods*,") as well as the negative ones between the alternative word and the weak word in the corresponding sentence (e.g., the alternative word "*tune*" and the weak word "*boy*" from the sentence "*the whistling boy was always out of tune.*") Finally, based on selected constraints, they used a large amount of parallel paraphrase data to train a seq2seq generator for rewriting the candidate sentence into a homophonic pun (e.g., "*the whistling fisherman was always out of tuna.*")

A supervised retrieval-based pipeline was introduced by Zhou et al. [146], which aims at transforming a given literal sentence into its idiomatic counterpart. The authors designed three different modules in their system: (i) a retrieval module that is used to retrieve the appropriate idiom for a given literal sentence from a pool of available idioms and their definitions; (ii) a span extraction module that identifies the span of the literal input to be replaced with the retrieved idiom; (iii) the generation module which is used to generate the idiomatic text based on the retrieved idiom and the literal input without the identified span.

RL for FLG. Researchers have also employed **reinforcement learning** (**RL**) techniques to optimize specific objectives in FLG. The main idea is to design a reward strategy that targets the desired figure of speech and employ the algorithms (i.e., policy gradient [137]) to maximize the expected reward during training. Zhu et al. [148] presented an unsupervised style transfer model to explore irony generation. They designed an irony reward strategy to control the irony accuracy and used a denoising auto-encoder and back-translation to preserve content. Additionally, they introduced a sentiment reward strategy to preserve sentiment polarity in the transformation from non-ironic to ironic. Luo et al. [77] presented an unsupervised framework for pun generation, which consists of a generator producing pun texts, and a pun classifier distinguishing between the generated text and the literal texts with specific word senses. Here, the probability output of the classifier is used as a reward signal to train the generator through reinforcement learning, which encourages the generator to produce texts that support two-word senses simultaneously. Mishra et al. [83] developed an unsupervised framework to generate sarcastic texts given literal negative opinions. Given that sarcasm comes from context-incongruity, the authors designed three modules to introduce incongruity into the literal input: (i) filtering factual content; (ii) retrieving incongruous phrases; and (iii) synthesising sarcastic text. The framework is trained with non-parallel sarcastic and literal texts through reinforced neural seq2seq learning, which takes the sarcastic confidence score of the discriminator (e.i. literal vs. sarcastic) as the reward for the figurative form.

In summary: on the one hand, training a neural model usually requires large-scale training data (parallel or non-parallel), to improve the model's generalization ability; also, the training process usually needs large computational resources. On the other hand, neural models can be trained based on the users' needs without too much extra-linguistic knowledge or complex linguistic processing, although integrating linguistic knowledge and retrieval approaches into neural networks can enhance the models' performance and interpretability.

4.2 Fine-Tuning Pre-trained Models

The development of PLMs is considered a revolutionary breakthrough in NLP and even neural networks. PLMs can be trained by self-supervised learning on a large amount of raw textual data, learning language knowledge and representations in a universal space. The models are then fine-tuned with usually limited amounts of labelled data for downstream NLP tasks using task-specific objective functions. Therefore, this paradigm can avoid training models from scratch and usually achieves state-of-the-art performance.

Context training. Similar to [116], Zhang and Wan [145] proposed a masking framework for hyperbole generation without literal-hyperbole pairs. They first fine-tuned the seq2seq PLM BART [65] to infill masked hyperbolic spans of source sentences in the training process, then masked part of the input literal sentence and fed it into the model to generate multiple candidate hyperbolic sentences during the inference process. Finally, a ranker was used to select the best candidate based on the hyperbolicity score and paraphrase quality of the sentences. Similarly, Zhou et al. [147] fine-tuned BART to generate a literal paraphrase given an idiomatic sentence. Specifically, they trained a masked conditional sentence generation model to fill the masked word using the definition of the word and its part-of-speech tag. Based on parallel data of simile and simile-free segment contexts, Zhang et al. [143] trained a framework consisting of BERT [27] based encoders and Transformers [128] based decoders for the task of writing polishment with similes. They first locate places in a given text where similes can be inserted and then generate location-specific similes from them (e.g., "*He appeared there [INSERT], holding the door frame with one hand, blocking her retreat*"→"*He appeared there like a ghost, holding the door frame with one hand, blocking her retreat.*")

Pseudo-parallel data construction and training. PLMs have been explored for automatically compiling parallel training data in FLG. Chakrabarty et al. [21] first collected comments containing the phrase *like a* from the social media site Reddit, and then leveraged the pre-trained generative model of the knowledge graph COMET [13] to transform simile into literal text utilizing the *PROPERTY* relation. This process creates literal-simile pairs, which are then used to fine-tune BART to generate novel similes for a given literal sentence. Stowe et al. [117] identified and masked metaphoric verbs of sentences from Gutenberg Poetry corpus [49], then employed BERT to infill the mask tokens. Based on the resulting large-scale parallel data, the authors proposed to control the generation process by encoding conceptual mappings between cognitive domains. Accordingly, they incorporated both target and source domains using FrameNet [6] into the source literal sentence as control codes. The pairs are then used to fine-tune BART to generate meaningful metaphoric texts. Similarly, Chakrabarty et al. [23] and Stowe et al. [115] employed masked language modelling combined with commonsense inferences to automatically create a large number of literal-metaphor pairs. Chakrabarty et al. [23] designed a metaphor detection model as a discriminator to guide the model's decoding during generation, addressing the problem that the model tends to generate literal tokens rather than metaphorical ones. In order to compare the free and controlled generation methods, Stowe et al. [115] fine-tuned the PLM T5 [101] for metaphor generation. For the free generation, the model is trained with original literal-metaphoric pairs while the controlled generation involves adding additional constraints to the generation objective, encouraging the model to generate a specific metaphor. They found that controlled generation can improve the metaphoricity of the generated sentences, while free generation tends to generate more fluent and coherent outputs.

Combining knowledge resource. Unlike Chakrabarty et al. [21] who used COMET to construct a parallel dataset, Tian et al. [123] leveraged it to develop a framework for hyperbole generation directly. They employed COMET and its reverse models to perform commonsense and counterfactual inference, and then generate multiple hyperbole candidates. Finally, they trained a generic hyperbole classifier and a specific pattern classifier to rank and select high-quality hyperboles. Chakrabarty et al. [19] also explored a retrieve-and-edit method using COMET. They developed an unsupervised framework for sarcasm generation given literal input sentences, which models two main characteristics of sarcasm: reversal of sentiment valence and semantic incongruity with the context. To implement the reversal of valence, they used WordNet [81] and SentiWordNet [30] to identify the evaluative word and select antonyms to generate the sarcastic utterance. Then, they employed COMET to retrieve relevant commonsense context to be added to the generated sarcastic texts. Mittal et al. [84] first used a reverse dictionary to generate a list of related concepts that are monosemous for both senses given two sense definitions of a target pun word. Based on the related concepts, they explored three methods: extractive-based, similarity-based, and generative-based, to generate context words. Finally, they fine-tuned T5 to generate humorous puns given the pun word and generated context words. Chakrabarty et al. [18] proposed the task of generating a plausible continuation given fictional narratives containing a figurative expression such as an idiom or a simile. To do so, they introduced a knowledge-enhanced strategy (PARA-COMET [35] and COMET-ConceptNET [48]) to fine-tune PLMs such as BART, T5, and GPT-2. This helps the models to infer the meaning from the context and rely on the literal meanings of constituent words, possibly following more closely human strategies in interpreting figurative language. Based on COMET, Ruan et al. [109] proposed an extraction-generation-ranking approach for image-sarcasm generation. Specifically, they first extracted diverse information from an image and used COMET to infer the consequence of the sentimental descriptive caption to generate candidate sarcastic texts. Then, they designed a ranking algorithm that considers image-text relation, sarcasticness, and grammaticality to select a final text from the candidate texts. Tian et al. [122] presented a framework for the task of generating puns. For a given pun word pair, the authors first retrieved a context word and a phrase from a large corpus based on Wikipedia and Gutenberg BookCorpus. Then, they fine-tuned the PLM GPT-2 model to learn the task of generating a sentence containing the input, including a keyword and a phrase. Finally, they trained a word-level label predictor based on linguistic attributes of puns, which is used to steer the model to generate puns.

More recently, Lai and Nissim [61] presented the task of multi-figurative language generation, addressing the limitations of existing works that focus on modelling single figurative forms separately, thereby possibly missing out on shared characteristics. Specifically, they provided a benchmark for the automatic generation of five common figurative forms by combining together and reusing existing resources, and proposed a BART-based framework with a mechanism for injecting information about the target figure of speech (or literal expression) into the encoder. This approach enables the transformation of sentences between different forms, including literal and figurative, without relying on parallel figurative-figurative sentence pairs.

In summary, this paradigm of fine-tuning general PLMs for the target downstream tasks, and more specifically in the case of FLG discussed here, usually achieves state-of-the-art results. Many works have also employed PLMs to construct synthetic parallel data, which can be used for fine-tuning PLMs and thereby mitigate the need for large-scale labelled data. However, fine-tuning PLMs with specific training objectives often requires a large computational capacity due to their large-scale sizes.

4.3 Prompt Learning

While fine-tuning PLMs often yields state-of-the-art results, this strategy does not work too well if only a handful of examples are available for the target task, which is very common in NLP. A new paradigm, prompt learning has been proposed for the scenario where the number of available examples is limited. This approach can utilize limited data as supervised information to rapidly generalize to the target task [135]. In particular, with the development of LLMs such as GPT-3, prompt learning has shown impressive performance when prompting the model with a small number of labelled examples, or even under zero examples conditions (i.e., zero-shot learning). Generally, the prompting method is designed to manipulate the model behaviour by prepending the task instruction, allowing LLMs to generate the desired output [16, 75, 99, 101]. For example, a source sentence from an English-to-Italian translation task can be reformulated as "*Translate English to Italian:* [*source English sentence*]."

In the field of FLG, Reif et al. [106] presented a prompting and augmented zero-shot learning method for metaphor generation, which frames the generation process as a sentence rewriting task. Concretely, they leveraged large PLMs to perform zero-shot learning, which requires only natural language instruction and an example from a relevant task (e.g., style transfer) without the need for model fine-tuning or examples specific to metaphor generation. Chakrabarty et al. [18] conducted zero-shot experiments on GPT-2 XL and GPT-3 to generate a coherent and contextually consistent next sentence for a given narrative containing a figurative expression. They let the models generate up to 20 tokens, stopping when an end-of-sentence token was generated. In addition, they performed a few-shot experiment in which they prompted GPT-3 with four task examples followed by a test input, to guide the model to generate corresponding outputs. However, the experiment results indicate that regardless of model size, language models struggled to perform well in zero-shot and few-shot settings when compared to fine-tuning PLMs such as BART. Additionally, all methods still lag significantly behind human performance levels. Recently, Mittal et al. [84] investigated GPT-3's few-shot learning to generate puns with context. Specifically, they provided two examples with GPT-3 to generate context words, and then generate puns incorporating context words from two different senses. Similar to [18], they found that fine-tuning T5 outperforms the GPT-3 with the few-shot method in terms of generating funny pun sentences. Bhavya et al. [12] studied the task of generating analogies by prompting LLM InstructGPT [90] in the zero-shot setting. They found that InstructGPT is effective at generating meaningful analogies, and the largest model can achieve human-level performance in generating analogies for given target concepts. However, the model is pretty sensitive to different prompts in various temperature settings.

In summary, these prompting methods require only a few labelled samples or no samples at all leveraging the prior knowledge of LLMs. However, the inference process for LLMs still demands significant computational resources compared to fine-tuning general PLMs which could be much smaller than LLMs. On the other hand, LLMs are sensitive to prompts for different tasks, and it is not clear how a prompt influences a model's performance or how to design an effective prompt for a given task.

5 EVALUATION

Evaluation is one of the most important parts of natural language generation, since it not only involves the assessment of the system, but also influences the modelling approach. For instance, while automatic metrics can be used to quickly evaluate different models, giving researchers the performance feedback they need right during development, most works also include human evaluation at the final stage as it can provide more reliable assessments. However, the evaluation in

Table 4. Automatic Metrics used for the Automatic Evaluation and Criteria Set for the Human Evaluation in FLG from 34 Papers

Automatic Evaluation				**Human Evaluation**			
Metric	**Total**	**Metric**	**Total**	**Criterion**	**Total**	**Criterion**	**Total**
BLEU	12	Average Length	3	Fluency	13	Success Rate	4
BERT(Score)	8	Novelty	2	Creativity	8	Adequacy	3
Perplexity	7	Embedding Similarity	2	Grammaticality	6	Funniness	3
Distinct	7	BLEURT	2	Coherence	5	Sarcasticness	2
ROUGE	6	Log-Likelihood	2	Metaphoricity	4	Semantic Similarity	2
Rhetoric Classifier	5	Word Number	2	Meaning	4	Readability	2
METEOR	4	HM/GM	2	Overall	4	Relevance	2
Others	18	—	—	Others	32	—	—

FLG, whether carried out by humans or through automated methods, continues to pose a challenge due to its subtle structures and subjective interpretations. Generally, evaluating a generated text is a multifaceted task requiring the assessment of different criteria, such as context preservation,[2] style strength, coherency, fluency, and so on.

5.1 Automatic Evaluation

Automatic metrics are very popular due to the ease of running automated assessments as systems are being developed, quickly providing a reproducible and scalable evaluation. Table 4 (left block) shows some commonly used metrics in the automatic evaluation of FLG, while let us bear in mind that most metrics are not designed specifically for FLG. These metrics can be divided into three different dimensions: (i) **context preservation**: whether the context of the generated sentence is the same as the original sentence; (ii) **form strength**: to what extent the generated text fits the target figurative form; and (iii) **fluency**: how fluent the language in the generated text is.

Context Preservation. The generated texts are compared to human-written references (or source texts) in terms of their semantic similarity using automatic metrics. As with other NLG tasks, there are many semantic metrics that can be applied to measure sentence similarity in FLG. These metrics can be categorized into surface-based and neural-based methods:

- BLEU [92] is the most popular metric used in many natural language generation tasks, including machine translation, question answering, text summarization, and style transfer. It compares generated texts to source or reference texts using a precision-oriented approach based on *n*-gram overlap. A BLEU-*n* score can be roughly computed as follows:

$$\text{BLEU-}n = \frac{\sum_{C \in \{Candidates\}} \sum_{n-gram \in C} Count_{match}(n-gram)}{\sum_{C \in \{Candidates\}} \sum_{n-gram \in C} Count(n-gram)}, \tag{6}$$

  where C represents a generated text in the candidate set, and *match* means that a *n*-gram appears in both the generated and reference texts.
- ROUGE [70] is another metric based on overlap counting, but a recall-oriented approach commonly used in automatic summarization evaluation. The formula of ROUGE-*n* is

[2] We follow previous work [21, 61, 123, 145] in using "context," instead of "content" which is commonly used in style transfer, suggesting that it is more the general context/topic of the source sentence that has to be preserved rather than its very exact content.

as follows:

$$\text{ROUGE-}n = \frac{\sum_{C\in\{Candidates\}} \sum_{n-gram\in C} Count_{match}(n-gram)}{\sum_{R\in\{References\}} \sum_{n-gram\in R} Count(n-gram)}, \tag{7}$$

where R represents a candidate text in the reference set. Compared to the BLEU formula in Equation (6), the only difference is the denominator which is the total number of n-grams in the reference set.

- METEOR [7] is an automatic metric based on a generalized concept of unigram matching between the generated texts and their corresponding references. It computes the similarity score of two texts using a combination of unigram-precision, unigram-recall, and some additional measures like stem and synonymy matching. Specifically, WordNet is used to expand the synonym set, and the word form is also taken into account for stem matching. The candidate can therefore be assessed even without an exact match with the references.
- Embedding similarity [73] measures the semantic distance between texts using pre-trained word embeddings, such as Word2Vec [80]. This kind of metric usually calculates the cosine distance between output and reference using sentence-level embeddings. *Embedding Average* measures a sentence's embeddings by averaging the word embeddings of each token in the sentence. *Vector Extrema* is another way to calculate sentence-level embeddings: it takes the most extreme value (maximum or minimum) amongst all word vectors for each dimension of the word vectors, and uses that value for sentence-level embedding. *Greedy matching*, instead, is a greedily token matching method based on the cosine similarity of word embeddings, and the overall score is then averaged across all words:

$$\text{G}(C,R) = \frac{\sum_{w_1\in C} \max_{w_2\in R} \text{cosine}(e_{w_1}, e_{w_2})}{|C|} \quad , \quad \text{GM}(C,R) = \frac{G(C,R)+G(R,C)}{2}.$$

  Where C and R represent the candidate output and reference, respectively. Since the formula G $(\cdot)$ is asymmetric, the greedy matching scores in both directions are averaged to get a final value.
- BERTScore [71] is a BERT-based automatic metric for text generation. Instead of an exact match, it uses greedy matching to compute a similarity score for each token in the candidate text with each token in the reference sentence, using contextual embeddings. This metric combines precision and recall to compute an F measure, which can be formulated as

$$\text{R} = \frac{1}{|C|}\sum_{c\in C}\max_{r\in R} \mathbf{c}^\top \mathbf{r} \quad , \quad P = \frac{1}{|R|}\sum_{r\in R}\max_{c\in C} \mathbf{c}^\top \mathbf{r} \quad , \quad F = 2\frac{R\cdot P}{R+P} \quad .$$

  Where C and R represent the candidate and reference, respectively.
- BLEURT [112] is a learnable metric based on BERT and trained on human judgements. It employs a pre-training scheme that uses millions of synthetic examples to help the model generalize: it has shown a high correlation with human judgement in the evaluation of machine translation [112] and formality transfer [60].

$$\text{BLEURT}(C,R) = \mathbf{W}\mathbf{v}_{[CLS]} + \mathbf{b}. \tag{8}$$

  Where $\mathbf{W}$ and $\mathbf{v}$ are the weight matrix and bias vector respectively. $\mathbf{v}_{[CLS]}$ is the representation for the special [CLS] token concatenated at the beginning with the candidate output C and reference R.

In our review, we observe that, as in most NLG tasks, using BLEU is the most popular strategy in FLG due to its advantages of light weight and fast computation. However, n-gram based metrics usually fail to recognise information beyond the lexical level, and BLEU has shown with poor correlation with human judgement in the evaluation of machine translation [8], question answering [17, 73], and formality transfer [15, 60]. In recent years, many neural network based metrics (e.g., BERTscore and BLEURT) have been developed as alternatives to n-gram based metrics. These metrics generally have a higher correlation with human judgement as they can capture semantic or syntactic variations of the given reference beyond the surface level.

Form Strength. The strength of the desired target form in the generated text can be assessed by a figurative language classifier, as commonly done with style classifier in style transfer [15, 60]. This process can be considered a binary classification problem (i.e., literal vs. figurative), where a classifier is used to automatically evaluate the form strength of the generated texts [76, 115]. A binary classifier, which has to be trained beforehand, is employed to assess whether each generated text adheres to the target figurative form. The recall, precision, and F score are then used to assess the model's performance. Research on text style transfer has shown that such classifiers correlate well with human judgement [60, 66].

Using a regressor in place of a classifier is another way to assess the strength of the figurative form, modelling the problem on a continuous scale of figurativeness rather than with discrete categories.

Fluency. This is a fundamental characteristic of human language that frequently plays an important role in the evaluation of NLG. To automate the assessment of this aspect, the **perplexity** (**PPL**) of generated texts is calculated against a language model that has been pre-trained with in-domain training data. This can be formulated as

$$\mathrm{PPL} = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{p(w_i|w_{1:i-1})}}, \tag{9}$$

where n is the number of tokens in the generated text, and $p(w_i)$ is the probability of the ith token w predicted by the language model. A low perplexity indicates high quality of the evaluated texts since it measures the probability of tokens appearing in text sequences.

From Table 4, we see that various automatic methods have been employed in FLG, including more than 10 metrics that are used only once. This suggests that the NLP community lacks shared standards in automatic evaluation, and it is yet unclear if these metrics are adequate for evaluating generated creative text. Extended research is needed specifically on automatic FLG evaluation.

5.2 Human Evaluation

Human evaluation can often provide a more reliable assessment of the FLG compared to automatic metrics although it is obviously more costly and can not be easily used during iterative development. To conduct a human evaluation, specific criteria aspects, such as context preservation, creativity, fluency, and overall judgement can be assessed. Annotators can rate the generated texts based on these aspects, or even compare and rank the outputs generated by different models at the same time.

Table 4 (right block) presents human evaluation criteria commonly used in FLG. Similar to automatic evaluation, we see that (i) the most commonly assessed criteria aspects are fluency, creativity, meaning, and overall; (ii) there are a variety of measures, with more than 30 criteria being used no more than twice again, suggesting a lack of agreement in strategy for human evaluation; (iii) more importantly, due to the lack of relevant research, there is no standard human evaluation framework that can be used to avoid biased assessment such as reference bias in machine translation evaluation [34], which might result in different works that are neither directly comparable nor easy to reproduce.

Table 5. A Summary of the Automatic Evaluation and Human Evaluation

Evaluation Method		Pros	Cons
Context preservation	BLEU	- Intuitive - Commonly used - Easy-to-implement	- Lexical-level evaluation
	ROUGE	- Intuitive - Commonly used - Easy-to-implement	- Lexical-level evaluation
	METEOR	- Intuitive - Commonly used - Easy-to-implement - Considering non-exact matching	- Lexical-level evaluation - Unigram matching only
	Embedding similarity	- Semantically measuring	- Lexical-level evaluation
	BERTScore	- Semantically measuring - Contextual learning	- Lexical-level evaluation - Large computational capacity
	BLEURT	- Semantically measuring - Mimicking human judgment	- Large computational capacity
Form Strength	Classifier	- Commonly used	- Having to train specific classifiers
Fluency	Perplexity	- Intuitive - Common	- Having to train specific LMs
Human evaluation	Human evaluation	- More reliable	- Costly - Time-consuming - Requiring high-quality annotators

Table 6. Benchmark Results for each FLG Task

Figure of Speech	Task	Dataset	Reference	Automatic Evaluation		Human Evaluation	
				Metric	Result	Metric	Result
Simile	Simile→Context with simile	[143]	[143]	Distinct	87.8	creativity (0,1)	0.7
Metaphor	Literal→Metaphor	[32]	[106]	–	–	Style Strength (1–100)	>75
Hyperbole	Literal→Hyperbole	[124]	[61]	Form Strength	84.4	–	–
Idiom	Literal→Idiom	[146]	[61]	Form Strength	76.4	–	–
Irony (Sarcasm)	Image→Sarcasm	[79]	[109]	Cosine Similarity	25.3	Sarcasticness (unknown)	2.9
Pun	Word Senses→Pun	[82]	[84]	Distinct-1	96.3	Fun (1–5)	3.0
Personification	Topic→Personification	[76]	[76]	BLEU	47.0	Rhetorical Aesthetics (1–5)	3.2

In Table 5, we summarize the pros and cons of the above-mentioned evaluation methods. Overall, there are not as yet fully shared standards in the practices of human-based and automatic evaluations.

5.3 Benchmark Results

Table 6 shows benchmark results for each figure of speech. It is worth noting that these results vary across tasks and evaluation methods and are therefore not comparable to each other. Here we aim at providing a simple glimpse of the current status of this field, including the state-of-the-art models (see references) and their corresponding evaluation. For automatic evaluation, we see that different models are evaluated with different metrics, making it difficult to derive a consistent overview. When looking at human evaluation, we observe that in all cases performances cannot be described as particularly good. These observations suggest that FLG (i) still needs plenty of work towards more performant models, and (ii) is in dire need of a shared evaluation framework.

6 CHALLENGES AND FUTURE DIRECTIONS

In spite of the recent interest in NLP, figurative language generation is still relatively understudied compared to other NLG tasks, though the progress of this field would be beneficial for a variety of practical applications, such as education. In this section, we outline several challenging problems and prospective research directions that we believe are critical and valuable, as well as some potential impacts of FLG.

6.1 FLG in Multi-Figurative/-Lingual Settings

Multi-Figurative Language Generation. At present, most existing work mainly focuses on modelling a single figurative form, i.e., reformulating a literal text into one with a specific target figure of speech. This strategy has the disadvantage of having to train separate models, one for each figure of speech, and of not exploiting potential knowledge transfer across figurative forms. Multi-figurative language generation is thus an interesting research direction. Recently, Lai and Nissim [61] took the first step towards multi-figurative language modelling and provided a benchmark for the automatic generation of five common figurative forms. This direction could potentially be further explored in the future, including (i) expanding and modelling more figures of speech; and (ii) exploring knowledge transfer across figurative forms, especially leveraging high-resource forms to model low-resource ones, since datasets are not equally available for the different figurative forms.

Multi-Lingual Figurative Language Generation. As discussed in Section 2.3, almost all existing FLG research focuses on English, with a small amount of work on Chinese and German. Apart from extending FLG work to other languages, looking at this task from a multilingual viewpoint might shed even more light on cross-lingual regularities and thus potentially help to tackle this task better both theoretically and from a practical perspective. A multilingual viewpoint in figurative language can also provide some insights into machine translation and thus advance this area. The main problem is the lack of data for other languages so that more work should be expected and welcomed in this direction. Cross-lingual learning can also support the learning of computational models for low-resource languages and domains.

Benchmark Datasets. Like many other tasks, using standard benchmark datasets is a crucial step in FLG research that can enable the development of various models, the assessment of their performances, and comparisons among them. Therefore, creating high-quality datasets is crucial and can greatly advance the progress of this field. In recent years, some (large) datasets of different figures of speech have been produced. However, most datasets are in English and in single figurative form (e.g., literal-to-idiom). As future work, more benchmark datasets developed in a multi-figurative and multilingual perspective as discussed above are needed and would also form a significant contribution. Datasets existing for yet other partially related tasks that have to do with creative language and double meanings, such as Tricky riddles [144], could also be considered in the context of FLG.

6.2 Methodology

We limit here our discussion to issues related to neural models as they are the most popular approach to FLG, and the one with the most potential for future development.

Reducing the need for parallel data. Fine-tuning PLMs can avoid having to train the model from scratch, thereby accelerating the convergence of the network. Although this recently has achieved some promising results in FLG, it still requires good amounts of task-specific parallel data (e.g., literal-idiom pairs). Self-supervised learning has been applied to FLG, for example masking the figurative words in the sentence and predicting which words should replace these masks [116, 145]. While these works just apply self-supervised learning for the specific downstream task, they also suggest a potential direction for training tailor-designed PLMs for FLG. This strategy of training a figurative-mask language model is worth exploring to reduce the need for task-specific parallel data in the fine-tuning stage. The key challenge is how to design an effective self-supervised training objective that is close to the target task. Another strategy is to exploit prior domain and task knowledge learned from transfer learning and pre-training to explore unsupervised approaches that do not require parallel data, such as PLM-based back translation and cross-figurative knowledge transfer.

RL for FLG. Although some existing works have employed reinforcement learning algorithms for FLG and achieved promising results, they focused on training neural models from scratch [77, 83, 148]. These works, however, have not been well studied when using PLMs. Therefore, one possible line of research is to augment PLMs with reward learning. As future work, we believe it is well worth exploring how to build reliable reward strategies to improve and balance multiple core aspects of FLG (e.g., form strength and context preservation), including metric-based reward methods [103], reinforcement learning with human feedback (RLHF, Ouyang et al. [90]), or direct preference optimization (DPO, Rafailov et al. [100]).

LLMs in FLG. The capabilities of LLMs have been continuously increasing by scaling model sizes, dataset sizes, and computation, with various preference optimization techniques [90, 100]. There are some works that have applied prompt learning with LLMs in FLG [18, 84, 106], where task instructions with few or no examples are designed to guide the model on the target task not always with great success. However, as introduced in Section 4.3, most work primarily focused on GPT-3, while more powerful models such as ChatGPT[3] and GPT-4 [88], have not been used for FLG. Additionally, we believe there is a need to explore prompt learning methods based on LLMs specifically for FLG. For instance, what mostly influences LLMs to control the figure of speech of the generated text? And how can prompts and instructions be designed to better control these variables? On the other hand, optimizing LLMs specifically for FLG has received less attention, which recently has been employed in a more general domain of creative writing [134].

6.3 Evaluation Methods

As discussed in Section 4, the NLP community has not yet reached fully shared standards and protocols for the evaluation of FLG. For instance, there are many automatic metrics, but the commonly used metrics are mainly borrowed from other NLG generation tasks (e.g., BLEU) and are not necessarily appropriate for this task. Furthermore, to the best of our knowledge, there is no work that focuses on evaluation practices, including the standard framework for conducting human evaluation alongside automatic metrics. Therefore, research on new evaluation metrics that target FLG is an important direction in the future. A good evaluation metric should be able to reflect the true performance of different models and can be used to guide researchers in developing and improving them. Most importantly, it should have a high correlation with human judgements. Ideally, new automatic metrics should be able to evaluate the relation between the source and output, that between reference and output, or among all of them. Therefore, more work on evaluation practices, that is, employing human judgement to navigate automatic metrics, either existing or new ones, is an interesting direction. In the context of LLM-based evaluation, recent work shows that ChatGPT achieves state-of-the-art or competitive correlation with golden human judgments on many NLG tasks by prompting the model with specific task institutions [62, 133]. This suggests that LLMs can also potentially be applied to the evaluation of FLG, which requires more exploration and validation.

[3] https://openai.com/blog/chatgpt.

As a final note, in the context of human evaluation of LLM's output, recent work [20] suggests to rely on the **Torrance Test metrics for Creative Thinking** (**TTCT**) for assessing LLMs' *creativity*. While there is not necessarily a correspondence between creativity and figurative language, this line of work could provide some future insights for extending and strengthening the human-based evaluation of FLG.

6.4 Applications

Based on the summary of current work, we can expect that the development of FLG not only affects other NLP tasks, but also can be applied to specific downstream applications or studies.

NLP Research. FLG can advance NLP downstream NLP tasks. For instance, the non-compositionality of idiom expressions exists in different degrees in different languages and is one of the open challenges in idiom translation [31]. The additional control of the figurative form of the translated text can be practically useful and also technically insightful. Also, computational approaches can be employed to provide a better understanding of linguistic phenomena and more specifically different figures of speech [61].

Conversational Agents. There are widely applied across different domains to serve various purposes, from providing automated assistance to companionship. Recent work shows that metaphors with various degrees of perceived warmth and competence can shape users' expectations of an agent, leading to different effects on such aspects, as willingness to use and cooperation [52]. Overall, we would expect that FLG can be a valuable complement to conversational agents, enhancing their ability to engage users in natural and expressive conversations, such as naturalness, emotional expression, personalization, and cultural sensitivity.

Education. Recent progress in generative AI has inspired substantial research on how humans could collaborate with LLMs for the creative writing [20, 134, 142]. Similarly, FLG can also be useful for creative writing assistance or even literary or poetic creation [21]. Therefore, educational applications can greatly benefit from FLG, for example in the form of automated modules that supply (second) language learning. Particularly, figurative language is an important part of K-12 education, LLMs-powered agents in FLG could be instrumental in creating diverse educational content, spanning introductions to literal and figurative language, advanced analysis, and writing exercises. Overall, we believe that figurative language generation can serve as a multifaceted tool for linguistic understanding, encompassing, including fostering comprehension, language proficiency, critical thinking, literary analysis, and communication skills.

6.5 Impact of FLG

In recent years, there has been increased attention to ethical issues related to Artificial Intelligence research. As with most language technologies, the development of FLG can benefit and improve human life with applications such as those discussed above. However, this could also lead to substantial inaccuracies, stereotypes, or demeaning, which could be propagated in further processing. For example, there is evidence of negative metaphors used in media discourse, which is a troubling potential harm [3]. With this in mind, a word of warning is necessary regarding the direct deployment of FLG models. Particularly, we should be wary of how this technology might be misused, and who might be harmed by it. Therefore, writing about risks explicitly in scientific papers advancing FLG research and also raising awareness of this possibility in the general public are ways to contain the effects of potentially harmful consequences. As practitioners in the field, we should be open to any discussion and suggestions to minimise such risks.

7 CONCLUSION

In this survey, we have comprehensively reviewed existing representative research work on figurative language generation, including common figures of speech, corresponding tasks, various approaches and evaluation strategies. Based on the critical analysis of the existing research trends, we have identified a series of key challenges and problems in this field and highlighted several directions for future work. We hope that this survey can provide researchers with a roadmap to easily track current research in FLG and grasp its core challenges, so as to make meaningful advances in this area.

REFERENCES

[1] Keiga Abe, Kayo Sakamoto, and Masanori Nakagawa. 2006. A computational model of the metaphor generation process. In *Proceedings of the 28th Annual Meeting of the Cognitive Science Society*. Retrieved from https://escholarship.org/uc/item/5d96219g

[2] Erez Lieberman Aiden and Jean-Baptiste Michel. 2011. Quantitative analysis of culture using millions of digitized books. *Science* 331, 6014 (2011), 176–182. DOI: https://doi.org/10.1126/science.1199644

[3] Liudmila Arcimaviciene and Sercan Hamza Bağlama. 2018. Migration, metaphor and myth in media representations: The ideological dichotomy of “Them” and “Us”. *SAGE Open* 8, 2 (2018), 1–13. Retrieved from https://api.semanticscholar.org/CorpusID:149710932

[4] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*. https://arxiv.org/abs/1409.0473

[5] Shuang Bai and Shan An. 2018. A survey on automatic image caption generation. *Neurocomputing* 311 (2018), 291–304. DOI: https://doi.org/10.1016/j.neucom.2018.05.080

[6] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*. Association for Computational Linguistics, Montreal, Quebec, Canada, 86–90. DOI: https://doi.org/10.3115/980845.980860

[7] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss (Eds.), Association for Computational Linguistics, Ann Arbor, Michigan, 65–72. Retrieved from https://aclanthology.org/W05-0909

[8] Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the 4th Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Marco Turchi, and Karin Verspoor (Eds.), Association for Computational Linguistics, Florence, Italy, 1–61. DOI: https://doi.org/10.18653/v1/W19-5301

[9] Valerio Basile and Johan Bos. 2011. Towards generating text from discourse representation structures. In *Proceedings of the 13th European Workshop on Natural Language Generation*. Claire Gardent and Kristina Striegnitz (Eds.), Association for Computational Linguistics, Nancy, France, 145–150. Retrieved from https://aclanthology.org/W11-2819

[10] Beata Beigman Klebanov, Chee Wee Leong, E. Dario Gutierrez, Ekaterina Shutova, and Michael Flor. 2016. Semantic classifications for detection of verb metaphors. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Berlin, Germany, 101–106. DOI: https://doi.org/10.18653/v1/P16-2017

[11] Beata Beigman Klebanov, Ekaterina Shutova, Patricia Lichtenstein, Smaranda Muresan, and Chee Wee (Eds.). 2018. In *Proceedings of the Workshop on Figurative Language Processing*. Association for Computational Linguistics, New Orleans, Louisiana. DOI: https://doi.org/10.18653/v1/W18-09

[12] Bhavya Bhavya, Jinjun Xiong, and ChengXiang Zhai. 2022. Analogy generation by prompting large language models: A case study of InstructGPT. In *Proceedings of the 15th International Conference on Natural Language Generation*. Samira Shaikh, Thiago Ferreira, and Amanda Stent (Eds.), Association for Computational Linguistics, Waterville, Maine, USA and virtual meeting, 298–312. DOI: https://doi.org/10.18653/v1/2022.inlg-main.25

[13] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Anna Korhonen, David Traum, and Lluís Màrquez (Eds.), Association for Computational Linguistics, Florence, Italy, 4762–4779. DOI: https://doi.org/10.18653/v1/P19-1470

[14] Andrea Bowes and Albert Katz. 2011. When sarcasm stings. *Discourse Processes* 48, 4 (2011), 215–236. DOI: https://doi.org/10.1080/0163853X.2010.532757 arXiv:https://doi.org/10.1080/0163853X.2010.532757

[15] Eleftheria Briakou, Sweta Agrawal, Joel Tetreault, and Marine Carpuat. 2021. Evaluating the evaluation metrics for style transfer: A case study in multilingual formality transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.), Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 1321–1336. DOI: https://doi.org/10.18653/v1/2021.emnlp-main.100

[16] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the Advances in Neural Information Processing Systems*. H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33, Curran Associates, Inc., 1877–1901. Retrieved from https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[17] Arun Chaganty, Stephen Mussmann, and Percy Liang. 2018. The price of debiasing automatic metrics in natural language evalaution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Iryna Gurevych and Yusuke Miyao (Eds.), Association for Computational Linguistics, Melbourne, Australia, 643–653. DOI: https://doi.org/10.18653/v1/P18-1060

[18] Tuhin Chakrabarty, Yejin Choi, and Vered Shwartz. 2022. It's not rocket science: Interpreting figurative language in narratives. *Transactions of the Association for Computational Linguistics* 10 (2022), 589–606. DOI: https://doi.org/10.1162/tacl_a_00478

[19] Tuhin Chakrabarty, Debanjan Ghosh, Smaranda Muresan, and Nanyun Peng. 2020. Rˆ3: Reverse, retrieve, and rank for sarcasm generation with commonsense knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.), Association for Computational Linguistics, Online, 7976–7986. DOI: https://doi.org/10.18653/v1/2020.acl-main.711

[20] Tuhin Chakrabarty, Philippe Laban, Divyansh Agarwal, Smaranda Muresan, and Chien-Sheng Wu. 2023. Art or artifice? Large language models and the false promise of creativity. arXiv:2309.14556. Retrieved from https://arxiv.org/abs/2309.14556

[21] Tuhin Chakrabarty, Smaranda Muresan, and Nanyun Peng. 2020. Generating similes effortlessly like a pro: A style transfer approach for simile generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.), Association for Computational Linguistics, Online, 6455–6469. DOI: https://doi.org/10.18653/v1/2020.emnlp-main.524

[22] Tuhin Chakrabarty, Arkadiy Saakyan, Debanjan Ghosh, and Smaranda Muresan. 2022. FLUTE: Figurative language understanding through textual explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.), Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 7139–7159. DOI: https://doi.org/10.18653/v1/2022.emnlp-main.481

[23] Tuhin Chakrabarty, Xurui Zhang, Smaranda Muresan, and Nanyun Peng. 2021. MERMAID: Metaphor generation with symbolism and discriminative decoding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.), Association for Computational Linguistics, Online, 4250–4261. DOI: https://doi.org/10.18653/v1/2021.naacl-main.336

[24] Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Kevin Knight, Ani Nenkova, and Owen Rambow (Eds.), Association for Computational Linguistics, San Diego, California, 93–98. DOI: https://doi.org/10.18653/v1/N16-1012

[25] Emilie Colin, Claire Gardent, Yassine M'rabet, Shashi Narayan, and Laura Perez-Beltrachini. 2016. The WebNLG challenge: Generating text from DBPedia data. In *Proceedings of the 9th International Natural Language Generation Conference*. Amy Isard, Verena Rieser, and Dimitra Gkatzia (Eds.), Association for Computational Linguistics, Edinburgh, UK, 163–167. DOI: https://doi.org/10.18653/v1/W16-6626

[26] Verna Dankers, Marek Rei, Martha Lewis, and Ekaterina Shutova. 2019. Modelling the interplay of metaphor and emotion through multitask learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, Hong Kong, China, 2218–2229. DOI: https://doi.org/10.18653/v1/D19-1227

[27] 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). Jill Burstein, Christy Doran, and Thamar Solorio (Eds.), Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. DOI: https://doi.org/10.18653/v1/N19-1423

[28] Ellen Dodge, Jisup Hong, and Elise Stickles. 2015. MetaNet: Deep semantic automatic metaphor analysis. In *Proceedings of the 3rd Workshop on Metaphor in NLP*. Ekaterina Shutova, Beata Beigman Klebanov, and Patricia Lichtenstein (Eds.), Association for Computational Linguistics, Denver, Colorado, 40–49. DOI: https://doi.org/10.3115/v1/W15-1405

[29] Abhijeet Dubey, Lakshya Kumar, Arpan Somani, Aditya Joshi, and Pushpak Bhattacharyya. 2019. "When Numbers Matter!": Detecting sarcasm in numerical portions of text. In *Proceedings of the 10th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Association for Computational Linguistics, Minneapolis, USA, 72–80. DOI: https://doi.org/10.18653/v1/W19-1309

[30] Andrea Esuli and Fabrizio Sebastiani. 2006. SENTIWORDNET: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*. Nicoletta Calzolari, Khalid Choukri, Aldo Gangemi, Bente Maegaard, Joseph Mariani, Jan Odijk, and Daniel Tapias (Eds.), European Language Resources Association (ELRA), Genoa, Italy. Retrieved from http://www.lrec-conf.org/proceedings/lrec2006/pdf/384_pdf.pdf

[31] Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2018. Examining the tip of the iceberg: A data set for idiom translation. In *Proceedings of the 11th International Conference on Language Resources and Evaluation*. European Language Resources Association (ELRA), Miyazaki, Japan. Retrieved from https://aclanthology.org/L18-1148

[32] Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Iryna Gurevych and Yusuke Miyao (Eds.), Association for Computational Linguistics, Melbourne, Australia, 889–898. DOI: https://doi.org/10.18653/v1/P18-1082

[33] Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 1615–1625. DOI: https://doi.org/10.18653/v1/D17-1169

[34] Marina Fomicheva and Lucia Specia. 2016. Reference bias in monolingual machine translation evaluation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Berlin, Germany, 77–82. DOI: https://doi.org/10.18653/v1/P16-2013

[35] Saadia Gabriel, Chandra Bhagavatula, Vered Shwartz, Ronan Le Bras, Maxwell Forbes, and Yejin Choi. 2021. Paragraph-level commonsense transformers with recurrent memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 12857–12865. Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/17521/17328

[36] Cristina Garbacea and Qiaozhu Mei. 2020. Neural language generation: Formulation, methods, and evaluation. arXiv:2007.15780. Retrieved from https://arxiv.org/abs/2007.15780

[37] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning* (Proceedings of Machine Learning Research, Vol. 70). Doina Precup and Yee Whye Teh (Eds.), PMLR, 1243–1252. Retrieved from https://proceedings.mlr.press/v70/gehring17a/gehring17a.pdf

[38] Katy Ilonka Gero and Lydia B. Chilton. 2019. Metaphoria: An algorithmic companion for metaphor creation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk). Association for Computing Machinery, New York, NY, USA, 1–12. Retrieved from https://dl.acm.org/doi/10.1145/3290605.3300526

[39] Debanjan Ghosh, Beata Beigman Klebanov, Smaranda Muresan, Anna Feldman, Soujanya Poria, and Tuhin Chakrabarty (Eds.). 2022. *Proceedings of the 3rd Workshop on Figurative Language Processing*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid). Retrieved from https://aclanthology.org/2022.flp-1.0

[40] Debanjan Ghosh, Elena Musi, and Smaranda Muresan. 2020. Interpreting verbal irony: Linguistic strategies and the connection to thetype of semantic incongruity. In *Proceedings of the Society for Computation in Linguistics 2020*. Association for Computational Linguistics, New York, New York, 82–93. Retrieved from https://aclanthology.org/2020.scil-1.10.pdf

[41] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. Retrieved from http://www.deeplearningbook.org

[42] Hessel Haagsma, Johan Bos, and Malvina Nissim. 2020. MAGPIE: A large corpus of potentially idiomatic expressions. In *Proceedings of the 12th Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 279–287. Retrieved from https://aclanthology.org/2020.lrec-1.35

[43] Sarah Harmon. 2015. FIGURE8: A novel system for generating and evaluating figurative language. In *Proceedings of the 6th International Conference on Computational Creativity*. 71–77. Retrieved from https://computationalcreativity.net/iccc2015/proceedings/4_1Harmon.pdf

[44] 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).* Jill Burstein, Christy Doran, and Thamar Solorio (Eds.), Association for Computational Linguistics, Minneapolis, Minnesota, 1734–1744. DOI: https://doi.org/10.18653/v1/N19-1172

[45] Raquel Hervás, Rui P. Costa, Hugo Costa, Pablo Gervás, and Francisco C. Pereira. 2007. Enrichment of automatically generated texts using metaphor. In *Proceedings of the Advances in Artificial Intelligence.* Retrieved from https://link.springer.com/chapter/10.1007/978-3-540-76631-5_90

[46] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 12(1997), 1735–80. Retrieved from https://www.bioinf.jku.at/publications/older/2604.pdf

[47] Bryan Anthony Hong and Ethel Ong. 2009. Automatically extracting word relationships as templates for pun generation. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity.* Anna Feldman and Birte Loenneker-Rodman (Eds.), Association for Computational Linguistics, Boulder, Colorado, 24–31. Retrieved from https://aclanthology.org/W09-2004

[48] Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence.* 6384–6392. Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/16792/16599

[49] Arthur M. Jacobs. 2018. The gutenberg english poetry corpus: Exemplary quantitative narrative analyses. *Frontiers Digit. Humanit.* 5 (2018), 5. Retrieved from https://www.frontiersin.org/articles/10.3389/fdigh.2018.00005/full

[50] Mark Alan Jones. 1992. Generating a specific class of metaphors. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, Newark, Delaware, USA, 321–323. DOI: https://doi.org/10.3115/981967.982019

[51] Aditya Joshi, Anoop Kunchukuttan, Pushpak Bhattacharyya, and Mark Carman. 2015. SarcasmBot: An open-source sarcasm-generation module for chatbots. In *Proceedings of the 4th International Workshop on Issues of Sentiment Discovery and Opinion Mining.* Retrieved from https://www.cse.iitb.ac.in/ adityaj/sarcasmbot-wisdom15-kdd.pdf

[52] Pranav Khadpe, Ranjay Krishna, Li Fei-Fei, Jeffrey T. Hancock, and Michael S. Bernstein. 2020. Conceptual metaphors impact perceptions of human-AI collaboration. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW2(2020), 26 pages. DOI: https://doi.org/10.1145/3415234

[53] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. 2017. Structured attention networks. In *5th International Conference on Learning Representations, Toulon, France, April 24–26, 2017, Conference Track Proceedings.* OpenReview.net. Retrieved from https://openreview.net/forum?id=HkE0Nvqlg

[54] Beata Beigman Klebanov, Ekaterina Shutova, Patricia Lichtenstein, Smaranda Muresan, Chee Wee, Anna Feldman, and Debanjan Ghosh (Eds.). 2020. *Proceedings of the 2nd Workshop on Figurative Language Processing.* Association for Computational Linguistics, Online. Retrieved from https://aclanthology.org/2020.figlang-1.0

[55] Philipp Koehn. 2010. *Statistical Machine Translation.* (1st. ed.) Cambridge University Press, USA.

[56] 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.* Association for Computational Linguistics, Online, 7024–7034. DOI: https://doi.org/10.18653/v1/2020.emnlp-main.571

[57] Ioannis Korkontzelos, Torsten Zesch, Fabio Massimo Zanzotto, and Chris Biemann. 2013. SemEval-2013 task 5: Evaluating phrasal semantics. In *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics, Volume 2: Proceedings of the 7th International Workshop on Semantic Evaluation*, Suresh Manandhar and Deniz Yuret (Eds.), Association for Computational Linguistics, Atlanta, Georgia, USA, 39–47. Retrieved from https://aclanthology.org/S13-2007

[58] Roger J. Kreuz and Richard M. Roberts. 1993. The empirical study of figurative language in literature. *Poetics* 22, 1 (1993), 151–169. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/0304422X9390026D

[59] Sachi Kumon-Nakamura, Sam Glucksberg, and M. Brown. 1995. How about another piece of pie: the allusional pretense theory of discourse irony. *Journal of Experimental Psychology. General* 124, 1 (1995), 3–21. Retrieved from https://psycnet.apa.org/record/1995-21193-001

[60] Huiyuan Lai, Jiali Mao, Antonio Toral, and Malvina Nissim. 2022. Human judgement as a compass to navigate automatic metrics for formality transfer. In *Proceedings of the 2nd Workshop on Human Evaluation of NLP Systems.* Association for Computational Linguistics, Dublin, Ireland, 102–115. DOI: https://doi.org/10.18653/v1/2022.humeval-1.9

[61] Huiyuan Lai and Malvina Nissim. 2022. Multi-figurative language generation. In *Proceedings of the 29th International Conference on Computational Linguistics.* International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 5939–5954. Retrieved from https://aclanthology.org/2022.coling-1.519

[62] Huiyuan Lai, Antonio Toral, and Malvina Nissim. 2023. Multidimensional evaluation for text style transfer using ChatGPT. arXiv:2304.13462. Retrieved from https://arxiv.org/abs/2304.13462

[63] Huiyuan Lai, Antonio Toral, and Malvina Nissim. 2023. Multilingual multi-figurative language detection. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, Toronto, Canada, 9254–9267. DOI: https://doi.org/10.18653/v1/2023.findings-acl.589

[64] John Leggitt and Raymond Gibbs. 2000. Emotional reactions to verbal irony. *Discourse Processes - DISCOURSE PROCESS* 29, 01(2000), 1–24. DOI: https://doi.org/10.1207/S15326950dp2901_1

[65] 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.* Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.), Association for Computational Linguistics, Online, 7871–7880. DOI: https://doi.org/10.18653/v1/2020.acl-main.703

[66] 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).* Association for Computational Linguistics, New Orleans, Louisiana, 1865–1874. DOI: https://doi.org/10.18653/v1/N18-1169

[67] Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Pretrained language model for text generation: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI'21)*, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 4492–4499. https://doi.org/10.24963/ijcai.2021/612

[68] Linlin Li and Caroline Sporleder. 2009. Classifier combination for contextual idiom detection without labelled data. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, 315–323. Retrieved from https://aclanthology.org/D09-1033

[69] Yucheng Li, Chenghua Lin, and Frank Guerin. 2022. Nominal metaphor generation with multitask learning. In *Proceedings of the 15th International Conference on Natural Language Generation.* Samira Shaikh, Thiago Ferreira, and Amanda Stent (Eds.), Association for Computational Linguistics, Waterville, Maine, USA and virtual meeting, 225–235. DOI: https://doi.org/10.18653/v1/2022.inlg-main.18

[70] Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Text Summarization Branches Out.* Association for Computational Linguistics, Barcelona, Spain, 74–81. Retrieved from https://aclanthology.org/W04-1013

[71] Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Text Summarization Branches Out.* Association for Computational Linguistics, Barcelona, Spain, 74–81. Retrieved from https://aclanthology.org/W04-1013

[72] Changsheng Liu and Rebecca Hwa. 2016. Phrasal substitution of idiomatic expressions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* Kevin Knight, Ani Nenkova, and Owen Rambow (Eds.), Association for Computational Linguistics, San Diego, California, 363–373. DOI: https://doi.org/10.18653/v1/N16-1040

[73] Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing.* Jian Su, Kevin Duh, and Xavier Carreras (Eds.), Association for Computational Linguistics, Austin, Texas, 2122–2132. DOI: https://doi.org/10.18653/v1/D16-1230

[74] Hugo Liu and Push Singh. 2004. ConceptNet–A practical commonsense reasoning tool-kit. *BT Technology Journal* 22, 06(2004), 211–226. DOI: https://doi.org/10.1023/B:BTTJ.0000047600.45421.6d

[75] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys* 55, 9(2023), 35 pages. DOI: https://doi.org/10.1145/3560815

[76] Zhiqiang Liu, Zuohui Fu, Jie Cao, Gerard de Melo, Yik-Cheung Tam, Cheng Niu, and Jie Zhou. 2019. Rhetorically controlled encoder-decoder for modern chinese poetry generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.* Anna Korhonen, David Traum, and Lluís Màrquez (Eds.), Association for Computational Linguistics, Florence, Italy, 1992–2001. DOI: https://doi.org/10.18653/v1/P19-1192

[77] Fuli Luo, Shunyao Li, Pengcheng Yang, Lei Li, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. 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.* Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.), Association for Computational Linguistics, Hong Kong, China, 3388–3393. DOI: https://doi.org/10.18653/v1/D19-1336

[78] Rui Mao, Chenghua Lin, and Frank Guerin. 2018. Word embedding and WordNet based metaphor identification and interpretation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1:*

Long Papers). Association for Computational Linguistics, Melbourne, Australia, 1222–1231. DOI: https://doi.org/10.18653/v1/P18-1113

[79] Alexander Mathews, Lexing Xie, and Xuming He. 2016. SentiCap: Generating image descriptions with sentiments. In *Proceedings of the 13th AAAI Conference on Artificial Intelligence* (Phoenix, Arizona). AAAI Press, 3574–3580. Retrieved from https://xmhe.bitbucket.io/papers/senti_desc_full.pdf

[80] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Advances in Neural Information Processing Systems*. C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger (Eds.), Vol. 26, Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf

[81] George A. Miller. 1995. WordNet: A lexical database for english. *Communications of the ACM* 38, 11(1995), 39–41. DOI: https://doi.org/10.1145/219717.219748

[82] Tristan Miller, Christian Hempelmann, and Iryna Gurevych. 2017. SemEval-2017 task 7: Detection and interpretation of english puns. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Steven Bethard, Marine Carpuat, Marianna Apidianaki, Saif M. Mohammad, Daniel Cer, and David Jurgens (Eds.), Association for Computational Linguistics, Vancouver, Canada, 58–68. DOI: https://doi.org/10.18653/v1/S17-2005

[83] Abhijit Mishra, Tarun Tater, and Karthik Sankaranarayanan. 2019. A modular architecture for unsupervised sarcasm generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.), Association for Computational Linguistics, Hong Kong, China, 6144–6154. DOI: https://doi.org/10.18653/v1/D19-1636

[84] Anirudh Mittal, Yufei Tian, and Nanyun Peng. 2022. AmbiPun: Generating humorous puns with ambiguous context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (Eds.), Association for Computational Linguistics, Seattle, United States, 1053–1062. DOI: https://doi.org/10.18653/v1/2022.naacl-main.77

[85] Saif Mohammad, Ekaterina Shutova, and Peter Turney. 2016. Metaphor as a medium for emotion: An empirical study. In *Proceedings of the 5th Joint Conference on Lexical and Computational Semantics*. Claire Gardent, Raffaella Bernardi, and Ivan Titov (Eds.), Association for Computational Linguistics, Berlin, Germany, 23–33. DOI: https://doi.org/10.18653/v1/S16-2003

[86] Malvina Nissim and Katja Markert. 2003. Syntactic features and word similarity for supervised metonymy resolution. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Sapporo, Japan, 56–63. DOI: https://doi.org/10.3115/1075096.1075104

[87] Geoffrey Nunberg, Ivan A. Sag, and Thomas Wasow. 1994. Idioms. *Language* 70, 3 (1994), 491–538. Retrieved from https://philpapers.org/rec/NUNI

[88] OpenAI. 2023. GPT-4 technical report. arXiv:2303.08774. Retrieved from https://arxiv.org/abs/2303.08774

[89] Dan Otter, Julian Richard Medina, and Jugal Kumar Kalita. 2021. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems* 32, 2 (2021), 604–624. Retrieved from https://ieeexplore.ieee.org/document/9075398

[90] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the Advances in Neural Information Processing Systems*. S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35, Curran Associates, Inc., 27730–27744. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a 73914f58805a001731-Paper-Conference.pdf

[91] Ekaterina Ovchinnikova, Vladimir Zaytsev, Suzanne Wertheim, and Ross Israel. 2014. Generating conceptual metaphors from proposition stores. arXiv:1409.7619. Retrieved from https://arxiv.org/abs/1409.7619

[92] 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*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 311–318. DOI: https://doi.org/10.3115/1073083.1073135

[93] Eric Partridge. 1999. *Usage & Abusage*. Penguin Books.

[94] Anthony M. Paul. 1970. Figurative language. *Philosophy and Rhetoric* 3, 4 (1970), 225–248. Retrieved from https://www.jstor.org/stable/40237206

[95] Lotem Peled and Roi Reichart. 2017. Sarcasm SIGN: Interpreting sarcasm with sentiment based monolingual machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Regina Barzilay and Min-Yen Kan (Eds.), Association for Computational Linguistics, Vancouver, Canada, 1690–1700. DOI: https://doi.org/10.18653/v1/P17-1155

[96] Francisco Pereira, Raquel Hervás, Pablo Gervás, and Amílcar Cardoso. 2006. A multiagent text generator with simple rhetorical habilities. *AAAI Workshop - Technical Report*(2006). Retrieved from http://nil.fdi.ucm.es/sites/default/files/CABH06.pdf

[97] Saša Petrović and David Matthews. 2013. Unsupervised joke generation from big data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers).* Hinrich Schuetze, Pascale Fung, and Massimo Poesio (Eds.), Association for Computational Linguistics, Sofia, Bulgaria, 228–232. Retrieved from https://aclanthology.org/P13-2041

[98] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. *OpenAI blog* 1–12.

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

[100] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Proceedings of the ICML 2023 Workshop The Many Facets of Preference-Based Learning.* Retrieved from https://openreview.net/forum?id=53HUHMvQLQ

[101] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67. Retrieved from https://jmlr.org/papers/volume21/20-074/20-074.pdf

[102] Harsh Rangwani, Devang Kulshreshtha, and Anil Kumar Singh. 2018. NLPRL-IITBHU at SemEval-2018 task 3: Combining linguistic features and emoji pre-trained CNN for irony detection in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation.* Association for Computational Linguistics, New Orleans, Louisiana, 638–642. DOI: https://doi.org/10.18653/v1/S18-1104

[103] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *Proceedings of the 4th International Conference on Learning Representations.* https://arxiv.org/abs/1511.06732

[104] Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality 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).* Association for Computational Linguistics, New Orleans, Louisiana, 129–140. DOI: https://doi.org/10.18653/v1/N18-1012

[105] Stefanie Regel. 2009. *The Comprehension of Figurative Language: Electrophysiological Evidence on the Processing of Irony.* Ph. D. Dissertation. Max Planck Institute for Human Cognitive and Brain Sciences Leipzig. Retrieved from https://pure.mpg.de/rest/items/item_726953/component/file_726952/content

[106] Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2022. A recipe for arbitrary text style transfer with large language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers).* Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.), Association for Computational Linguistics, Dublin, Ireland, 837–848. DOI: https://doi.org/10.18653/v1/2022.acl-short.94

[107] Richard M. Roberts and Roger J. Kreuz. 1994. Why do people use figurative language? *Psychological Science* 5, 3 (1994), 159–163. Retrieved from https://journals.sagepub.com/doi/10.1111/j.1467-9280.1994.tb00653.x

[108] Omid Rohanian, Shiva Taslimipoor, Richard Evans, and Ruslan Mitkov. 2018. WLV at SemEval-2018 task 3: Dissecting tweets in search of irony. In *Proceedings of The 12th International Workshop on Semantic Evaluation.* Association for Computational Linguistics, New Orleans, Louisiana, 553–559. DOI: https://doi.org/10.18653/v1/S18-1090

[109] Jie Ruan, Yue Wu, Xiaojun Wan, and Yuesheng Zhu. 2024. Describe images in a boring way: Towards cross-modal sarcasm generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.* 5701–5710. Retrieved from https://openaccess.thecvf.com/content/WACV2024/papers/Ruan_Describe_Images_in_a_Boring_Way_Towards_Cross-Modal_Sarcasm_Generation_WACV_2024_paper.pdf

[110] Giancarlo Salton, Robert Ross, and John Kelleher. 2014. An empirical study of the impact of idioms on phrase based statistical machine translation of english to brazilian-portuguese. In *Proceedings of the 3rd Workshop on Hybrid Approaches to Machine Translation.* Rafael E. Banchs, Marta R. Costa-jussà, Reinhard Rapp, Patrik Lambert, Kurt Eberle, and Bogdan Babych (Eds.), Association for Computational Linguistics, Gothenburg, Sweden, 36–41. DOI: https://doi.org/10.3115/v1/W14-1007

[111] John Schwoebel, Shelly Dews, Ellen Winner, and Kavitha Srinivas. 2000. Obligatory processing of the literal meaning of ironic utterances: Further evidence. *Metaphor and Symbol* 15, 1-2 (2000), 47–61. Retrieved from https://www.tandfonline.com/doi/abs/10.1080/10926488.2000.9678864

[112] Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, Online, 7881–7892. DOI: https://doi.org/10.18653/v1/2020.acl-main.704

[113] Ekaterina Shutova. 2010. Automatic metaphor interpretation as a paraphrasing task. In *Proceedings of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for*

Computational Linguistics. Ron Kaplan, Jill Burstein, Mary Harper, and Gerald Penn (Eds.), Association for Computational Linguistics, Los Angeles, California, 1029–1037. Retrieved from https://aclanthology.org/N10-1147

[114] Ekaterina V. Shutova. 2011. Computational approaches to figurative language. *Technical Report* 803 (2011) Retrieved from https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-803.pdf

[115] Kevin Stowe, Nils Beck, and Iryna Gurevych. 2021. Exploring metaphoric paraphrase generation. In *Proceedings of the 25th Conference on Computational Natural Language Learning.* Arianna Bisazza and Omri Abend (Eds.), Association for Computational Linguistics, Online, 323–336. DOI: https://doi.org/10.18653/v1/2021.conll-1.26

[116] Kevin Stowe, Nils Beck, and Iryna Gurevych. 2021. Exploring metaphoric paraphrase generation. In *Proceedings of the 25th Conference on Computational Natural Language Learning.* Arianna Bisazza and Omri Abend (Eds.), Association for Computational Linguistics, Online, 323–336. DOI: https://doi.org/10.18653/v1/2021.conll-1.26

[117] Kevin Stowe, Tuhin Chakrabarty, Nanyun Peng, Smaranda Muresan, and Iryna Gurevych. 2021. Metaphor generation with conceptual mappings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers).* Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.), Association for Computational Linguistics, Online, 6724–6736. DOI: https://doi.org/10.18653/v1/2021.acl-long.524

[118] Kevin Stowe, Prasetya Utama, and Iryna Gurevych. 2022. IMPLI: Investigating NLI models' performance on figurative language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.), Association for Computational Linguistics, Dublin, Ireland, 5375–5388. DOI: https://doi.org/10.18653/v1/2022.acl-long.369

[119] Jiao Sun, Anjali Narayan-Chen, Shereen Oraby, Shuyang Gao, Tagyoung Chung, Jing Huang, Yang Liu, and Nanyun Peng. 2022. Context-situated pun generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing.* Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.), Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 4635–4648. DOI: https://doi.org/10.18653/v1/2022.emnlp-main.306

[120] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* (Montreal, Canada). MIT Press, Cambridge, MA, USA, 3104–3112. Retrieved from https://proceedings.neurips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf

[121] Asuka Terai and Masanori Nakagawa. 2010. A computational system of metaphor generation with evaluation mechanism. In *Proceedings of the 20th International Conference on Artificial Neural Networks.* 142–147. Retrieved from https://link.springer.com/chapter/10.1007/978-3-642-15822-3_18

[122] Yufei Tian, Divyanshu Sheth, and Nanyun Peng. 2022. A unified framework for pun generation with humor principles. In *Findings of the Association for Computational Linguistics: EMNLP 2022.* Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.), Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 3253–3261. DOI: https://doi.org/10.18653/v1/2022.findings-emnlp.237

[123] 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.* Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.), Association for Computational Linguistics, Punta Cana, Dominican Republic, 1583–1593. DOI: https://doi.org/10.18653/v1/2021.findings-emnlp.136

[124] 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.* Association for Computational Linguistics, Brussels, Belgium, 3296–3304. DOI: https://doi.org/10.18653/v1/D18-1367

[125] Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Association for Computational Linguistics, Baltimore, Maryland, 248–258. DOI: https://doi.org/10.3115/v1/P14-1024

[126] Alessandro Valitutti, Hannu Toivonen, Antoine Doucet, and Jukka M. Toivanen. 2013. "Let everything turn well in your wife": Generation of adult humor using lexical constraints. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers).* Hinrich Schuetze, Pascale Fung, and Massimo Poesio (Eds.), Association for Computational Linguistics, Sofia, Bulgaria, 243–248. Retrieved from https://aclanthology.org/P13-2044

[127] Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. SemEval-2018 task 3: Irony detection in english tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation.* Association for Computational Linguistics, New Orleans, Louisiana, 39–50. DOI: https://doi.org/10.18653/v1/S18-1005

[128] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems.* I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett

(Eds.), Vol. 30, Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[129] Tony Veale. 2016. Round up the usual suspects: Knowledge-based metaphor generation. In *Proceedings of the 4th Workshop on Metaphor in NLP*. Beata Beigman Klebanov, Ekaterina Shutova, and Patricia Lichtenstein (Eds.), Association for Computational Linguistics, San Diego, California, 34–41. DOI: https://doi.org/10.18653/v1/W16-1105

[130] Tony Veale and Yanfen Hao. 2008. A fluid knowledge representation for understanding and generating creative metaphors. In *Proceedings of the 22nd International Conference on Computational Linguistics*. Donia Scott and Hans Uszkoreit (Eds.), Coling 2008 Organizing Committee, Manchester, UK, 945–952. Retrieved from https://aclanthology.org/C08-1119

[131] Oriol Vinyals and Quoc Le. 2015. A neural conversational model. arXiv:1506.05869. Retrieved from https://arxiv.org/abs/1506.05869

[132] Martin Volk and Nico Weber. 1998. The automatic translation of idioms. machine translation vs. translation memory systems. *Sprachwissenschaft, Computerlinguistik und neue Medien*1 1 (1998) 167–192. Retrieved from https://www.zora.uzh.ch/id/eprint/19070/

[133] Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is ChatGPT a good NLG evaluator? A preliminary study. In *Proceedings of the 4th New Frontiers in Summarization Workshop*. Yue Dong, Wen Xiao, Lu Wang, Fei Liu, and Giuseppe Carenini (Eds.), Association for Computational Linguistics, Singapore, 1–11. DOI: https://doi.org/10.18653/v1/2023.newsum-1.1

[134] Tiannan Wang, Jiamin Chen, Qingrui Jia, Shuai Wang, Ruoyu Fang, Huilin Wang, Zhaowei Gao, Chunzhao Xie, Chuou Xu, Jihong Dai, Yibin Liu, Jialong Wu, Shengwei Ding, Long Li, Zhiwei Huang, Xinle Deng, Teng Yu, Gangan Ma, Han Xiao, Zixin Chen, Danjun Xiang, Yunxia Wang, Yuanyuan Zhu, Yi Xiao, Jing Wang, Yiru Wang, Siran Ding, Jiayang Huang, Jiayi Xu, Yilihamu Tayier, Zhenyu Hu, Yuan Gao, Chengfeng Zheng, Yueshu Ye, Yihang Li, Lei Wan, Xinyue Jiang, Yujie Wang, Siyu Cheng, Zhule Song, Xiangru Tang, Xiaohua Xu, Ningyu Zhang, Huajun Chen, Yuchen Eleanor Jiang, and Wangchunshu Zhou. 2024. Weaver: Foundation models for creative writing. arXiv:2401.17268. Retrieved from https://arxiv.org/abs/2401.17268

[135] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys* 53, 3(2020), 34 pages. DOI: https://doi.org/10.1145/3386252

[136] Lowri Williams, Christian Bannister, Michael Arribas-Ayllon, Alun Preece, and Irena Spasić. 2015. The role of idioms in sentiment analysis. *Expert Systems with Applications* 42, 21 (2015), 7375–7385. Retrieved from https://www.sciencedirect.com/science/article/pii/S0957417415003759

[137] Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8 (1992), 229–256. Retrieved from https://link.springer.com/article/10.1007/BF00992696

[138] Chuhan Wu, Fangzhao Wu, Sixing Wu, Junxin Liu, Zhigang Yuan, and Yongfeng Huang. 2018. THU_NGN at SemEval-2018 task 3: Tweet irony detection with densely connected LSTM and multi-task learning. In *Proceedings of the 12th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, New Orleans, Louisiana, 51–56. DOI: https://doi.org/10.18653/v1/S18-1006

[139] 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)*. Iryna Gurevych and Yusuke Miyao (Eds.), Association for Computational Linguistics, Melbourne, Australia, 1650–1660. DOI: https://doi.org/10.18653/v1/P18-1153

[140] 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)*. Jill Burstein, Christy Doran, and Thamar Solorio (Eds.), Association for Computational Linguistics, Minneapolis, Minnesota, 861–871. DOI: https://doi.org/10.18653/v1/N19-1092

[141] 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*. Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.), Association for Computational Linguistics, Online, 2870–2876. DOI: https://doi.org/10.18653/v1/2020.emnlp-main.229

[142] Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. 2022. Wordcraft: Story writing with large language models. In *Proceedings of the 27th International Conference on Intelligent User Interfaces* (Helsinki, Finland). Association for Computing Machinery, New York, NY, USA, 841–852. DOI: https://doi.org/10.1145/3490099.3511105

[143] 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*. 14383–14392. Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/17691/17498

[144] Yunxiang Zhang and Xiaojun Wan. 2021. BiRdQA: A bilingual dataset for question answering on tricky riddles. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*. Retrieved from https://api.semanticscholar.org/CorpusID:237605111

[145] Yunxiang Zhang and Xiaojun Wan. 2022. MOVER: Mask, over-generate and rank for hyperbole generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 6018–6030. DOI: https://doi.org/10.18653/v1/2022.naacl-main.440

[146] Jianing Zhou, Hongyu Gong, and Suma Bhat. 2021. From solving a problem boldly to cutting the gordian knot: Idiomatic text generation. arXiv:2104.06541. Retrieved from https://arxiv.org/abs/2104.06541

[147] Jianing Zhou, Ziheng Zeng, Hongyu Gong, and Suma Bhat. 2022. Idiomatic expression paraphrasing without strong supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 11774–11782. DOI: https://doi.org/10.1609/aaai.v36i10.21433

[148] Mengdi Zhu, Zhiwei Yu, and Xiaojun Wan. 2019. A neural approach to irony generation. arXiv:1909.06200. Retrieved from https://arxiv.org/abs/1909.06200

Received 14 April 2023; revised 27 February 2024; accepted 10 March 2024