A Survey of Pun Generation: Datasets, Evaluations and Methodologies

Anonymous ACL submission

Abstract

Pun generation seeks to creatively modify linguistic elements in text to produce humour or evoke double meanings. It also aims to preserve coherence and contextual appropriateness, making it useful in creative writing and entertainment across various media and contexts. This field has been widely studied in computational linguistics, while there are currently no surveys that specifically focus on pun generation. To bridge this gap, this paper provides a comprehensive review of pun generation datasets and methods across different stages, including traditional approaches, deep learning techniques, and pre-trained language models. Additionally, we summarise both automated and human evaluation metrics used to assess the quality of pun generation. Finally, we discuss the research challenges and propose promising directions for future work.

1 Introduction

Pun is a kind of rhetorical style that leverages the polysemy or phonetic similarity of words to produce expressions with double or multiple meanings (Delabastita, 2016). Beyond mere wordplay, puns serve as a crucial mechanism of linguistic creativity, enriching communication and making it more engaging (Carter, 2015). For example, the pun sentence "I used to be a banker, but I lost interest" plays on the pun words "interest", encompassing both a lack of enthusiasm for banking as a profession and the idea of financial loss. This ability to encode multiple layers of meaning fosters cognitive flexibility, encouraging individuals to interpret language in innovative ways (Zheng and Wang, 2023). Due to the unique capacity of puns, they are widely used in advertising (Djafarova, 2008; Van Mulken et al., 2005), literature (Giorgadze, 2014), and various other fields.

Natural language generation (NLG) tasks involve the creation of human-like text by computers based on given data or input (Gatt and Krahmer, 2018), with pun generation being a notable and challenging aspect of such tasks. There are various approaches utilised in automatic pun generation, including template-based methods (Hong and Ong, 2009), deep neural network approaches (He et al., 2019), and pre-trained language models (PLMs) employing various training and inference styles (Mittal et al., 2022; Xu et al., 2024). These methods are applied to different types of puns, with a particular focus on homophonic (Yu et al., 2020), homographic (Yu et al., 2018; Luo et al., 2019), heterographic puns (Xu et al., 2024) and visual puns (Rebrii et al., 2022).

Despite the long-standing research interest in pun generation, a comprehensive literature review in this field has not been conducted, to the best of our knowledge. Some existing relevant surveys focus on generating creative writing and delve into tasks such as poetry composition (Bena and Kalita, 2020; Elzohbi and Zhao, 2023), storytelling (Gieseke et al., 2021; Alhussain and Azmi, 2021), arts (Shahriar, 2022) and metaphor (Rai and Chakraverty, 2020; Ge et al., 2023). It is noteworthy that Amin and Burghardt (2020) outlined methodologies to humour generation, discussing various systems based on templates and neural networks, along with their respective strengths and weaknesses. However, they did not cover the pun research nor incorporate relevant technologies associated with large language models (LLMs). Therefore, we aim to address this gap by conducting the first comprehensive survey on pun generation.

In this survey, we review the past three decades of research and examine the current state of natural language pun generation, categorising these methods in five groups based on their technological development timeline: (1) Traditional methods, which involve generating puns by manually or automatically constructing templates; (2) Classic Deep Neural networks (DNNs), leveraging architectures,

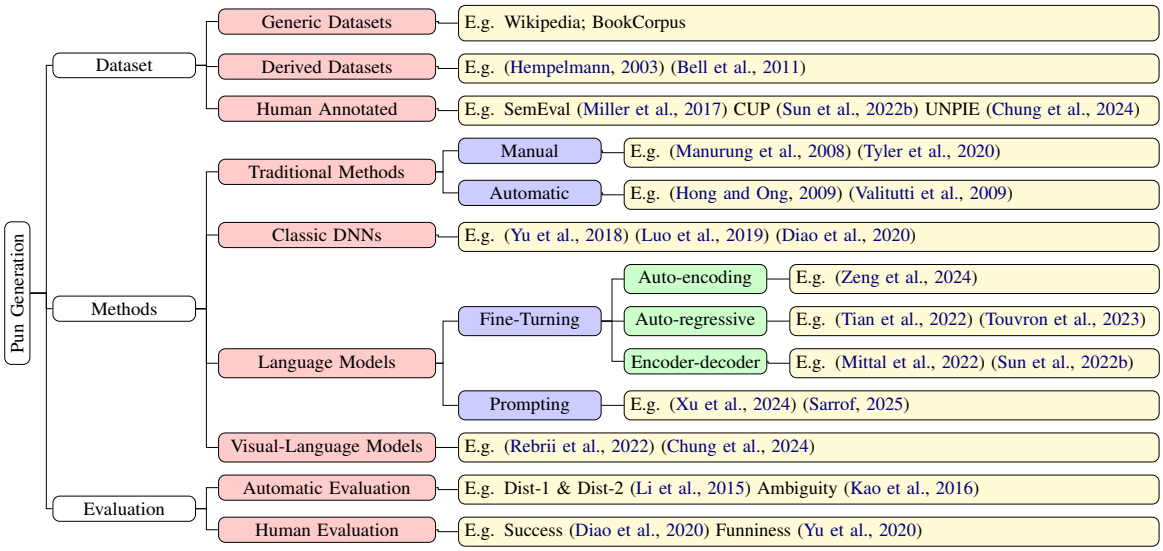

Figure 1: The survey tree for pun generation.

such as RNNs and their variants, to learn pun patterns from data; (3) Fine-tuning of PLMs, where pre-trained models like GPT (Radford, 2018) are adapted with task-specific datasets to improve pun generation, (4) Prompting of PLMs, which utilizes carefully designed prompts to guide models in generating puns without additional training, and (5) Visual-language models, where some preliminary studies on visual pun generation. We further summarise the automatic and human evaluation metrics used to assess the quality of generated puns. Finally, we discuss our findings and propose promising research directions for future work in this field.

The paper is organised as follows: Section 2 reviews the main categories of puns and provides examples for each category. Section 3, 4 and 5 summarise the relevant datasets, methods, and evaluation metrics, as shown in figure 1. We also discuss the challenges and outline future research directions in Section 6, as well as conclude with final remarks in Section 7.

2 Pun Categories

This section outlines the main four types of puns: i) *Homophonic puns*, ii) *Heterographic puns*, iii) *Homographic puns* and iv) *Visual pun*. The main features of these categories are listed in the Appendix A.

2.1 Homophonic Puns

Homophonic puns rely on the dual meanings of homophones, which are words that sound alike but have different meanings (Attardo, 2009), illustrated in example (a):

(a) Dentists don't like a hard day at the orifice (office).

which uses the "orifice" as the pivotal pun word. The term "orifice" refers to the human mouth, while its pronunciation is similar to "office". This similarity allows it to be interpreted as a dentist working in an office, thereby creating a humorous pun effect.

2.2 Heterographic Puns

Heterographic puns emphasize on differences in spelling with the same pronunciation to achieve their rhetorical effect, which are also classified as homophonic puns in some studies (Sun et al., 2022b; Miller et al., 2017). An example of a heterographic pun is shown as (b):

(b) Life is a puzzle, look here for the missing peace (piece). (Xu et al., 2024)

The word "peace" can be interpreted as tranquility in life, while it shares the same pronunciation as "piece" which refers to a puzzle piece. Therefore, the pun can be recognized as seeking either peace in life or the missing piece of a puzzle.

2.3 Homographic Puns

Homographic puns exploit words spelled the same homographs but possess different meanings (Attardo, 2009), as shown in example (c):

(c) Always trust a glue salesman. They tend to stick to their word.

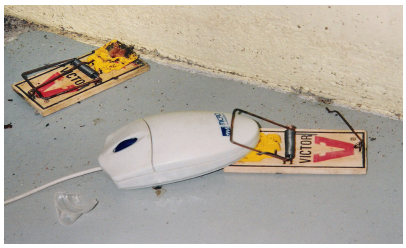

Figure 2: A visual pun example features a white mouse and a mouse trap, where the combination exploits the double meaning of the word “mouse”.

The phrase “stick to their word” refers to the act of keeping a promise in common English expressions. However, the meaning of “stick” is also directly associated with the adhesive properties of “glue”, which artfully plays on the dual meanings of the word “stick”.

2.4 Visual Puns

Visual puns are a form of artistic expression that utilize images or visual elements to create double meanings (Smith et al., 2008). A typical example of a visual pun from Wikipedia [1] is shown in Figure 2. The figure leverages the multiple meanings of the word "mouse" based on the computer device and animal, thereby creating a pun effect by combining the computer mouse and mousetrap.

3 Dataset

In this section, we present the current datasets that have been used and constructed for pun research. We classified the datasets into generic datasets, derived datasets and human-annotated datasets. For the detailed table of the pun dataset, please refer to Appendix C.

3.1 Generic Datasets

In the early days of neural network technology, due to the difficulty of obtaining adequate data to train seq2seq models for some specific tasks (Yu et al., 2018), most research in pun generation relied on general datasets to train conditional language models, enabling them to capture fundamental semantic relationships. For example, some pun generation studies use the English Wikipedia corpus to train the language model (Yu et al., 2018; Luo et al., 2019; Diao et al., 2020), while others rely on BookCorpus (Zhu, 2015; Yu et al., 2020) as a generic corpus for retrieval and training. Sarrof (2025) analysed the distribution of Hindi words in Latin and Devanagari scripts using C4 (Raffel et al., 2020) and The Pile (Gao et al., 2020), and then tested on the Dakshina dataset (Roark et al., 2020).

[1] https://en.wikipedia.org/wiki/Pun

3.2 Derived Datasets

The derived datasets are created for the new datasets by processing, transforming, or extracting specific details from general data. In this section, we present a list of derived datasets and outline the domains used in their creation. Sobkowiak (1991) collected 3850 puns from advertisements and conversation, while Hempelmann (2003) selected a subset for the automatic generation of heterophonic puns. Lucas (2004) proposed a tiny pun corpus that relies on lexical ambiguity from newspaper comics. Bell et al. (2011) created a 373 puns dataset from church marquees and literature to study wordplay in religious advertising. In addition, several studies have created pun datasets by filtering data from specialised joke websites. For example, both Yang et al. (2015) and Kao et al. (2016) curated pun datasets by crawling data from the "Pun of the Day" website. Jaech et al. (2016) compiled a homophonic pun dataset from Tumblr, Reddit, and Twitter to facilitate the automatic recovery of the target word in given puns.

3.3 Human Annotated

This section provides some details of human-annotated pun datasets. **SemEval.** Miller et al. (2017) released two manually annotated pun datasets based on (Miller and Turković, 2016) and (Miller, 2016) including both homophonic and heterogeneous puns, which is one of the most commonly used datasets in the pun generation community. **SemEval Enhancements.** Sun et al. (2022b) augmented the SemEval dataset by adding pun data combined with a given context and provided annotations on the adaptation between context words and their corresponding pun pairs. Furthermore, Sun et al. (2022a) added the fine-grained funniness ratings and natural language explanations based on the SemEval dataset. **ChinesePun.** Chen et al. (2024) introduced the first datasets for Chinese homophonic and homographic puns, specifically designed for pun understanding and generation tasks. **Multimodal Dataset.** Zhang et al. (2024) compiled a large collection of Chinese historical visual puns and provided detailed annotations, including the identification of prominent visual elements, matching of these elements with their symbolic meanings and interpretations. Chung et al. (2024)

selected a subset of homophonic and heterogeneous puns from the SemEval dataset and supplemented it with corresponding explanation images.

4 Methodology

In this section, we provide an overview of existing approaches to pun generation.

4.1 Traditional Models

Early traditional methods are typically through template-based construction. In linguistics, a template refers to a textual structure consisting of predefined slots that can be populated with various variables (Amin and Burghardt, 2020). Binsted and Ritchie (1994) developed the simple question-answer system of pun-generator Joke Analysis and Production Engine (JAPE), which was improved in subsequent versions including JAPE-2 (Binsted, 1996) and JAPE-3. The model incorporates two primary structures: schemata, which are used to explore the relationships between different keywords, and templates, which are designed to generate the basic framework for puns. Inspired by JAPE, Manurung et al. (2008) designed the STANDUP system, which expands and variants the elements generated by puns through further semantic and phonological analysis, for children with complex communication needs. Furthermore, Tyler et al. (2020) expanded upon the JAPE system by incorporating more recent knowledge bases and designed the PAUL BOT system, enhancing its capabilities and flexibility in automated pun generation.

Additionally, HCPP (Venour, 2000) and WISCRAIC (McKay, 2002) systems both implement models for the specific subclass of puns about homonym common phrase and idiom-based witticisms according to semantic associations, respectively. Hempelmann (2003) studies target recoverability, arguing that a robust model for target alternative words recovery provides the necessary foundation for heterographic pun generation. Ritchie (2005) considered pun generation from the broader perspective of NLG. They analyse the differences in mechanisms between pun generation and traditional NLG, as well as the computational methods that could potentially accomplish this task. As for the research on non-English puns, Dybala et al. (2008) designed a Japanese pun generator as part of a conversational system, while Dehouck and Delaborde (2025) proposed a generator for automatically generating French puns based on a given name and a word or phrase using rules.

Since building templates manually is a tedious and time-consuming task, Hong and Ong (2009) proposed Template-Based Pun Extractor and Generator (T-PEG) automatically identify, extract and represent the word relationships in a template, and then use these templates as patterns for the computer to generate its own puns. Valitutti et al. (2009) generated funny puns by implementing GraphLaugh to automatically generate different types of lexical associations and visualize them through a dynamic graph. They also explored a method for automatically generating humour through the substitution of words in short texts (Valitutti et al., 2013).

4.2 Classic DNNs

With the development of deep learning, pun generation has increasingly been implemented using deep neural networks, including Sequence-to-Sequence (Seq2Seq) (Sutskever, 2014) and Generative Adversarial Network (GAN) (Goodfellow et al., 2014). In general, Seq2Seq models map input sequences, such as words and phrases, to output the pun sentence, by maximising the conditional log-likelihood of the generated sequence.

Yu et al. (2018) represented the first attempt to apply deep neural networks to generate homographic puns without specific training data by developing a conditional language model (Mou et al., 2015) that creates sentences containing a target word with dual meanings. Building on this generator, Luo et al. (2019) introduced a novel discriminator, which is a word sense classifier with a single-layer bi-directional LSTM, to provide a well-structured ambiguity reward for the generator. Diao et al. (2020) replaced the traditional LSTM network structure with ON-LSTM (Shen et al., 2018) to further enhance performance. Additionally, He et al. (2019) and Yu et al. (2020) used the Seq2Seq model to rewrite the sentence so that it remains grammatically correct after replacing pun words.

In general, classic DNNs can generate puns that are more flexible compared to traditional models by fitting both general and pun datasets. However, existing methods heavily rely on annotated data and limited types of corpora, which restricts further improvement in the quality of pun generation.

4.3 Pre-trained Language Models

Early PLMs, such as Word2Vec (Mikolov, 2013) and GloVe (Pennington et al., 2014), are distributed

Method	Model	Type	Language	Dataset
Classic Deep Neural Networks				
Neural Pun (Yu et al., 2018)	LSTM	hog	English	Wikipedia & (Miller et al., 2017)
Pun-GAN (Luo et al., 2019)	LSTM	hog	English	Wikipedia & (Miller et al., 2017)
SurGen (He et al., 2019)	LSTM	hop	English	BookCorpus & (Miller et al., 2017)
LCR (Yu et al., 2020)	LSTM	hop	English	BookCorpus & (Hu et al., 2019)
AFPun-GAN (Diao et al., 2020)	ON-LSTM	hog	English	Wikipedia & (Miller et al., 2017)
Pre-trained Language Models				
Ext Ambipun(Mittal et al., 2022)	T5	hog	English	(Annamoradnejad and Zoghi, 2020)
Sim Ambipun(Mittal et al., 2022)	T5	hog	English	(Annamoradnejad and Zoghi, 2020)
Gen Ambipun(Mittal et al., 2022)	T5	hog	English	(Annamoradnejad and Zoghi, 2020)
UnifiedPun(Tian et al., 2022)	GPT-2 & BERT	hog&hog	English	(Annamoradnejad and Zoghi, 2020)
Context-pun(Sun et al., 2022b)	T5	hog&heg	English	(Sun et al., 2022b)
PunIntended (Zeng et al., 2024)	BERT	hop&hog	English	(Sun et al., 2022a)
PGCL (Chen et al., 2024)	LLaMA2-7B	hop&hog	English	(Miller et al., 2017)
PGCL (Chen et al., 2024)	Baichuan2-7B	hop&hog	Chinese	(Chen et al., 2024)
Hinglish (Sarrof, 2025)	GPT-3.5	hop	Multi-language	C4 & The Pile & Dakshina

Table 1: Methods of neural network models and pre-trained language models for pun generation task. Hog, hop and heg denote the types of homographic puns, homophonic puns and heterographic puns, respectively.

word representation methods trained on large-scale unlabeled text data, capable of capturing both the semantic and contextual information of words. These models are utilised to address various subtasks involved in pun generation. For example, Mittal et al. (2022) proposed to get the context words from Word2Vec based on pun words. Yu et al. (2020) designed a constraint selection algorithm based on lexical semantic relevance and obtained the word embeddings from Continuous Bag of Words model (CBOW) (Mikolov, 2013).

Most contemporary PLMs are built upon the Transformer architecture (Vaswani, 2017), which has shown outstanding performance across various natural language processing tasks (Min et al., 2023). The main model categories are classified into: (1) auto-encoding models, such as BERT (Devlin et al., 2019), (2) auto-regressive models, such as the GPT-2 (Radford et al., 2019), and (3) encoder-decoder models, such as T5 (Raffel et al., 2020). Pun generation tasks are primarily implemented through fine-tuning and prompting strategies.

4.3.1 PLMs with Fine-Tuning

Fine-tuning PLMs is to further train the model on a specific dataset to make it better suited to the needs of a specific task. For auto-encoding models, since the bidirectional encoding characteristics of the model are not suitable for generation tasks, most current work on pun generation employs it as the discriminator in GANs. For example, Zeng et al. (2024) and Tian et al. (2022) both used the BERT-base model, leveraging the [CLS] token representation for classification.

In auto-regressive models, Tian et al. (2022) fine-tuned the GPT-2 model based on the combination dataset of Gutenberg BookCorpus and jokes (Annamoradnejad and Zoghi, 2020) and proposed a unified framework for generating both homophonic and homographic puns. Chen et al. (2024) fine-tuned both LLaMA2-7B (Touvron et al., 2023) and Baichuan2-7B (Yang et al., 2023) for generating English and Chinese puns respectively through the standard Direct Preference Optimization (Rafailov et al., 2024) and multistage curriculum learning framework.

For encoder-decoder models, Mittal et al. (2022) explored the generation of puns based on context words associated with pun words and finetuned a keyword-to-sentence model using the T5 model. Similarly, Sun et al. (2022b) proposed the context-situated pun generation, which involves identifying pun words for a given set of contextual keywords and then generating puns based on these keywords and the associated pun words. Zeng et al. (2024) used T5 as a generator, taking the pun semantic trees as input and generating pun text as output.

4.3.2 PLMs with Prompting

Prompting (Liu et al., 2021) refers to a specially designed input mode intended to guide PLMs, especially for LLMs, in performing specific tasks (Alhazmi et al., 2024). However, there are few studies exploring pun generation specifically from the perspective of prompting. Mittal et al. (2022) provides examples of the target pun along with its

two interpretations and instructions for generating the pun in GPT-3 (Brown et al., 2020) to serve as a baseline comparison model. Based on the Chain-of-Thought prompting approach (Wei et al., 2022), Sarrof (2025) designed a novel method that integrates homophone and transliteration modules to enhance the quality of pun generation.

In addition, Xu et al. (2024) selected a range of prominent LLMs to evaluate their capabilities on pun generation, including both open-source models in Llama2-7B-Chat (Touvron et al., 2023), Mistral-7B (Jiang et al., 2023), Vicuna-7B (Zheng et al., 2023), and OpenChat-7B (Wang et al., 2023), and closed-source models in Gemini-Pro (Google, 2023), GPT-3.5-Turbo (OpenAI, 2023a), Claude3-Opus (Anthropic, 2024), and GPT-4-Turbo (OpenAI, 2023b). These studies reveal that although LLMs still exhibit limitations in generating creative and humorous puns, their demonstrated potential highlights a developmental trend in this field. Future research can further optimize existing LLMs to enhance their performance in pun generation tasks.

4.4 Visual-Language Models

There are currently some preliminary studies on visual puns. Rebrii et al. (2022) explored the cross-lingual translation of puns combined with visual elements. Chung et al. (2024) employed the DALLE-3 (Betker et al., 2023) to generate images that illustrated the meanings of puns based on textual puns. Zhang et al. (2024) leveraged their established dataset to conduct a comprehensive evaluation of large vision-language models in visual pun comprehension. However, to the best of our knowledge, there are no dedicated studies on visual pun generation, which is a potential future research direction.

5 Evaluation Strategies

In this section, we examine both automatic and human evaluation methods for pun generation. Table 2 summarizes the primary metrics for evaluation.

5.1 Automatic Evaluation

The automatic evaluation metrics can be categorized based on the intention and definition. We classify the metrics into funniness, diversity and fluency.

5.1.1 Funniness

Ambiguity & Distinctiveness. Kao et al. (2016) introduced the metrics of *ambiguity* and *distinctiveness* based on information theory. These metrics integrate computational models of general language understanding and pun features to quantitatively predict humour with fine-grained precision (Kao et al., 2016). Specifically, ambiguity refers to the uncertainty arising from multiple possible meanings within a sentence, which is formulated as:

$$Amb(M) = -\sum_{k\in\{a,b\}} P(m_k \mid \vec{w}) \log P(m_k \mid \vec{w}) \tag{1}$$

where $\vec{w}$ is a vector of observed content words in a sentence and m_k is the latent sentence meaning. Higher ambiguity allows the sentence to better support both the pun and its alternative meanings.

Distinctiveness evaluates the differences between word sets that support distinct meanings within a sentence using the symmetrized Kullback-Leibler divergence D_{KL}, defined as follows:

$$Dist(F_a, F_b) = D_{KL}(F_a \| F_b) + D_{KL}(F_b \| F_a) \tag{2}$$

where F_a and F_b represent the set of words in a sentence that support two different meanings along with their probability distributions. The high distinctiveness indicates that the distributions of the two-word groups differ significantly, which enhances the sense of humour.

Surprisal. Surprisal is a quantitative metric for surprise based on the pun word and the alternative word given local and global contexts (He et al., 2019). The formulation of local surprisal and global surprisal are defined as follows:

$$\begin{aligned} S_{\text{local}} &= S(x_{p-d:p-1}, x_{p+1:p+d}), \\ S_{\text{global}} &= S(x_{1:p-1}, x_{p+1:n}), \end{aligned} \tag{3}$$

where S is the log-likelihood ratio of two events, $x_1, \ldots, x_n$ is a sequence of tokens, p is the pun word and d is the local window size. Finally, a unified metric is defined as a ratio of local-global surprisal to quantify the success of the pun generation. Some details of the formulas are provided in the Appendix B.

5.1.2 Diversity

Unusualness. Given the uniqueness of puns, *unusualness* measures based on the normalised log probabilities from language models are also utilised for pun evaluation (He et al., 2019; Pauls and Klein, 2012), which is formulated as follows:

$$Unusualness \stackrel{def}{=} -\frac{1}{n} \log \left(\frac{p(x_1, \ldots, x_n)}{\prod_{i=1}^{n} p(x_i)} \right) \tag{4}$$

Paper	Automatic Evaluation							Human Evaluation					
	PPLs.	D1&2.	Succ.	Ambi.	Dist	Surp.	Unus.	Succ.	Funn.	Flun.	Info.	Cohe.	Read.
(Yu et al., 2018)	✓	✓	×	×	×	×	×	×	×	✓	×	✓	✓
(He et al., 2019)	×	×	×	✓	✓	✓	✓	✓	✓	×	×	×	×
(Luo et al., 2019)	×	✓	×	×	×	×	✓	✓	×	✓	×	×	×
(Yu et al., 2020)	×	✓	×	×	×	×	×	✓	✓	✓	×	×	×
(Diao et al., 2020)	×	✓	×	×	×	×	✓	✓	×	✓	×	×	×
(Mittal et al., 2022)	×	✓	×	×	×	×	×	✓	✓	×	×	✓	×
(Tian et al., 2022)	×	×	×	✓	✓	✓	×	✓	✓	×	✓	×	×
(Sun et al., 2022b)	×	×	✓	×	×	×	×	✓	×	×	×	×	×
(Zeng et al., 2024)	×	✓	×	✓	×	×	×	-	-	-	-	-	-
(Chen et al., 2024)	×	✓	✓	×	×	×	×	✓	×	×	×	×	×

Table 2: Main methods for automatic and human evaluation of pun generation. PPLs., D1&2., Succ., Ambi., Dist., Surp., and Unus. denote the metrics of Perplexity Score, Dist-1 & Dist-2, Structure Succ., Ambiguity, Distinctiveness, Surprisal, and Unusualness, respectively. Similarly, Succ., Funn., Gram., Flun., Info., Cohe., and Read. represent Success, Funniness, Grammar, Fluency, Informativeness, Coherence, and Readability. ✓ indicates metrics that are used, while × indicates metrics that are not used. The symbol "-" signifies that the method is not applicable to this evaluation.

where $p(x_1, \ldots, x_n)$ and $p(x_i)$ are the joint and independent probabilities, respectively. A higher metric result suggests the presence of uncommon collocations, innovative sentence structures, and other linguistic features, aligning with the characteristics of puns.

Dist-1 & Dist-2. Dist-1 and Dist-2 focus on the diversity of words and phrases in the generated text (Li et al., 2015), which calculates the proportion of unique n-grams to the total number of n-grams, as formulated Dist-1, for example:

$$\text{Dist-1} = \frac{unique\ unigrams}{total\ generated\ words} \tag{5}$$

where a higher Dist-1 score indicates greater diversity in the generated sentences, whereas a lower score suggests more generic and repetitive text. The DIst-2 formulation is shown as Appendix B.

5.1.3 Fluency

Perplexity score (Jelinek et al., 1977). This score evaluates whether the generated puns are natural and fluent. In practice, some studies (Yu et al., 2018) quantified by using the generative language model, formally described as follows:

$$perplexity = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(x_i|x_{<i})\right) \tag{6}$$

where $P(x_i|x_{<i})$ is the probability of the i-th token of a pun, given the sequence of tokens ahead.

Structure Succ. The evaluation measures the rate of contextual word and pun word integration, specifically the proportion of successful inclusion of pun words in the generated puns, formally shown as follows:

$$Succ = \frac{t_{correct}}{T} \times 100\% \tag{7}$$

where $t_{correct}$ is the number of generated puns with correctly included pun words and T is the total number of generated puns.

5.2 Human Evaluation

In the task of pun generation, since puns are a creative form of language (Yu et al., 2020), human evaluation is essential and intuitively assesses the quality of the generated puns. The primary evaluation metrics are: **Success** recognises whether the generated sentence qualifies as a successful pun based on the definition from (Miller et al., 2017); **Funniness** evaluates the humour and comedic quality of the generated sentences; **Fluency** shows whether the sentence is grammatically correct and flows naturally; **Informativeness** rates whether the generated sentences effectively convey meaningful and specific information; **Coherence** indicates whether the generated sentence is coherency and given senses are suitable in a sentence; **Readability** indicates whether the sentence is easy to understand semantically.

Most studies utilize the Likert Scale (Joshi et al., 2015) to assess the metrics. This commonly used psychological measurement method and relies numerical scales within a specific range to evaluate a given objective (Alhazmi et al., 2024). For example, Mittal et al. (2022) utilized a Likert scale ranging from 1 (not at all) to 5 (extremely) to rate the funniness and coherence of puns. In particular, for

success metrics, some studies adopt a binary classification method in which evaluators determine whether the generated pun is successful by selecting *True* or *False* (Tian et al., 2022; Sun et al., 2022b; Chen et al., 2024).

With the development of LLMs, Chen et al. (2024) conducted a human A/B test, asking annotators to compare paired puns generated by their methods and ChatGPT and select more humorous puns. Since GPT-4's evaluations aligned closely with those of human reviewers (Liang et al., 2024), Zeng et al. (2024) replaced human reviewers with GPT-4 to assess the metrics of readability, funniness, and coherence.

6 Challenges and Future Directions

This section outlines the challenges and explores potential directions for future work.

6.1 Multilingual Research

With advancements in pun generation research, the majority of studies focus primarily on English, as shown in Table 1, while studies on puns in other languages remain limited. Linguistically, different languages employ distinct mechanisms to create puns. For example, ideographic or mixed languages, such as Chinese and Japanese, tend to emphasize multi-layered pun construction (Shao et al., 2013). Therefore, cross-language pun generation can also serve as a potential future work. Building on previous cross-linguistic research, using parallel data, including word-parallel (Zhao et al., 2020; Alqahtani et al., 2021) and sentence-parallel (Reimers and Gurevych, 2020; Heffernan et al., 2022), can be utilized to achieve targeted alignment of pun words. Additionally, some pioneering works can capture phonological and semantic puns through advanced learning approaches such as contrastive learning (Hu et al., 2024), modify pre-training schemes (Clark, 2020) and adapter tuning (Pfeiffer et al., 2020; Parović et al., 2022).

6.2 Multi-Modal Information

Multimodal information enables a more reliable understanding of the world (Stein, 1993), and incorporating multiple modalities into tasks can enhance the quality of pun generation. Although previous studies have introduced some multimodal evaluations and datasets (Zhang et al., 2024; Chung et al., 2024), few have specifically focused on the generation of multimodal puns. One potential method is shared representation (Ngiam et al., 2011), which involves integrating complementary information from different modalities to learn higher-performance representations (Lahat et al., 2015). For example, automatic speech recognition (Malik et al., 2021) can be leveraged to enhance homophonic puns. Another direction is to translate puns between modalities, i.e., cross-modal generation (Suzuki and Matsuo, 2022), including text-to-image (Zhang et al., 2023a), image-to-text (He and Deng, 2017), text-to-speech (Zhang et al., 2023b) and speech-to-text (Shadiev et al., 2014; Fortuna and Nunes, 2018)

6.3 PLMs Prompting Design

While prompt engineering has proven effective in enhancing text generation capabilities of LLMs (Liu et al., 2023), current research still faces significant limitations in generating puns, such as an over-reliance on overly simplistic or single-faceted prompts. Chain-of-thought prompting is a powerful technique that significantly improves the reasoning capabilities of LLMs (Wei et al., 2022). Therefore, the quality of pun generation can be enhanced by transferring CoT technique from other fields, such as using iterative bootstrapping (Sun et al., 2023), knowledge enhancement (Dhuliawala et al., 2023; He et al., 2024), question decomposition (Trivedi et al., 2022) and self-ensemble (Yin et al., 2024). Furthermore, the resut can be improved by optimizing CoT's prompt construction, encompassing semi-automatic prompting (Shum et al., 2023) and automatic prompting (Zhang et al., 2022), as well as exploring diverse topological variants (Chu et al., 2024), such as chain structures (Olausson et al., 2023), tree structures (Ning et al., 2023), and graph structures (Besta et al., 2024).

7 Conclusion

In this paper, we present the first comprehensive survey on pun generation tasks, including phonetic, graphic and visual puns. We classify and conduct a thorough analysis of the datasets used in pun research, review previous approaches to pun generation, discuss existing methods, as well as summarize the evaluation metrics for pun generation. Furthermore, we highlight the challenges and future directions, offering valuable insights for researchers interested in pun generation. To enhance research in pun generation, this paper also plans to provide a continuously updated reading list available on a GitHub repository.

Limitations

Although we have attempted to extensively analyse the existing literature on pun generation, some works may still be missed due to variations in search keywords. Furthermore, our exploration of other categories of puns is limited, such as recursive puns and antanaclasis, as we encountered challenges while searching for them, which may be influenced by the relatively low attention they have received in the research community. Finally, due to the rapid development of the research field, this study does not cover the entire historical scope nor the latest advancements following the survey. However, our work represents the first comprehensive survey on pun generation, including datasets, methods, evaluation, challenges and potential directions, making it a valuable resource for scholars in this field.

References

Elaf Alhazmi, Quan Z. Sheng, W. Zhang, Munazza Zaib, and Ahoud Abdulrahmn F. Alhazmi. 2024. Distractor generation in multiple-choice tasks: A survey of methods, datasets, and evaluation. In *Conference on Empirical Methods in Natural Language Processing*.

Arwa I Alhussain and Aqil M Azmi. 2021. Automatic story generation: A survey of approaches. *ACM Computing Surveys (CSUR)*, 54(5):1–38.

Sawsan Alqahtani, Garima Lalwani, Yi Zhang, Salvatore Romeo, and Saab Mansour. 2021. Using optimal transport as alignment objective for fine-tuning multilingual contextualized embeddings. *arXiv preprint arXiv:2110.02887*.

Miriam Amin and Manuel Burghardt. 2020. A survey on approaches to computational humor generation. In *Proceedings of the 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 29–41.

Issa Annamoradnejad and Gohar Zoghi. 2020. Colbert: Using bert sentence embedding in parallel neural networks for computational humor. *arXiv preprint arXiv:2004.12765*.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku.

Salvatore Attardo. 2009. *Linguistic theories of humor*. Walter de Gruyter.

Nancy D Bell, Scott Crossley, and Christian F Hempelmann. 2011. Wordplay in church marquees.

Brendan Bena and Jugal Kalita. 2020. Introducing aspects of creativity in automatic poetry generation. *arXiv preprint arXiv:2002.02511*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. 2023. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8.

Kim Binsted. 1996. Machine humour: An implemented model of puns.

Kim Binsted and Graeme Ritchie. 1994. *An implemented model of punning riddles*. University of Edinburgh, Department of Artificial Intelligence.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Ma teusz 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. *ArXiv*, abs/2005.14165.

Ronald Carter. 2015. *Language and creativity: The art of common talk*. Routledge.

Yang Chen, Chong Yang, Tu Hu, Xinhao Chen, Man Lan, Li Cai, Xinlin Zhuang, Xuan Lin, Xin Lu, and Aimin Zhou. 2024. Are u a joke master? pun generation via multi-stage curriculum learning towards a humor llm. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 878–890.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. 2024. Navigate through enigmatic labyrinth a survey of chain of thought reasoning: Advances, frontiers and future. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1173–1203.

Jiwan Chung, Seungwon Lim, Jaehyun Jeon, Seungbeen Lee, and Youngjae Yu. 2024. Can visual language models resolve textual ambiguity with visual cues? let visual puns tell you! *arXiv preprint arXiv:2410.01023*.

K Clark. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Mathieu Dehouck and Marine Delaborde. 2025. Rule-based approaches to the automatic generation of puns based on given names in french. In *Proceedings of*

the 1st Workshop on Computational Humor (CHum), pages 18–22.

Dirk Delabastita. 2016. *Traductio: Essays on punning and translation*. Routledge.

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

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*.

Yufeng Diao, Liang Yang, Xiaochao Fan, Yonghe Chu, Di Wu, Shaowu Zhang, and Hongfei Lin. 2020. Afpun-gan: Ambiguity-fluency generative adversarial network for pun generation. In *Natural Language Processing and Chinese Computing: 9th CCF International Conference, NLPCC 2020, Zhengzhou, China, October 14–18, 2020, Proceedings, Part I 9*, pages 604–616. Springer.

Elmira Djafarova. 2008. Why do advertisers use puns? a linguistic perspective. *Journal of Advertising Research*, 48(2):267–275.

Pawel Dybala, Michal Ptaszynski, Shinsuke Higuchi, Rafal Rzepka, and Kenji Araki. 2008. Humor prevails!-implementing a joke generator into a conversational system. In *AI 2008: Advances in Artificial Intelligence: 21st Australasian Joint Conference on Artificial Intelligence Auckland, New Zealand, December 1-5, 2008. Proceedings 21*, pages 214–225. Springer.

Mohamad Elzohbi and Richard Zhao. 2023. Creative data generation: A review focusing on text and poetry. *arXiv preprint arXiv:2305.08493*.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30.

Vaishali Ganganwar, Manvainder, Mohit Singh, Priyank Patil, and Saurabh Joshi. 2024. Sarcasm and humor detection in code-mixed hindi data: A survey. In *International Conference on Computing and Machine Learning*, pages 453–469. Springer.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Mengshi Ge, Rui Mao, and Erik Cambria. 2023. A survey on computational metaphor processing techniques: From identification, interpretation, generation to application. *Artificial Intelligence Review*, 56(Suppl 2):1829–1895.

Lena Gieseke, Paul Asente, Radomír Měch, Bedrich Benes, and Martin Fuchs. 2021. A survey of control mechanisms for creative pattern generation. In *Computer Graphics Forum*, volume 40, pages 585–609. Wiley Online Library.

Meri Giorgadze. 2014. Linguistic features of pun, its typology and classification. *European Scientific Journal*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Gemini Team Google. 2023. Gemini: A Family of Highly Capable Multimodal Models. *arXiv e-prints*, arXiv:2312.11805.

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

Xiaodong He and Li Deng. 2017. Deep learning for image-to-text generation: A technical overview. *IEEE Signal Processing Magazine*, 34(6):109–116.

Zhiwei He, Tian Liang, Wenxiang Jiao, Zhuosheng Zhang, Yujiu Yang, Rui Wang, Zhaopeng Tu, Shuming Shi, and Xing Wang. 2024. Exploring human-like translation strategy with large language models. *Transactions of the Association for Computational Linguistics*, 12:229–246.

Kevin Heffernan, Onur Çelebi, and Holger Schwenk. 2022. Bitext mining using distilled sentence representations for low-resource languages. *arXiv preprint arXiv:2205.12654*.

Christian F Hempelmann. 2003. *Paronomasic puns: Target recoverability towards automatic generation*. Ph.D. thesis, Purdue University.

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*, pages 24–31.

Haigen Hu, Xiaoyuan Wang, Yan Zhang, Qi Chen, and Qiu Guan. 2024. A comprehensive survey on contrastive learning. *Neurocomputing*, page 128645.

J Edward Hu, Rachel Rudinger, Matt Post, and Benjamin Van Durme. 2019. Parabank: Monolingual bitext generation and sentential paraphrasing via lexically-constrained neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6521–6528.

Aaron Jaech, Rik Koncel-Kedziorski, and Mari Ostendorf. 2016. Phonological pun-derstanding. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 654–663.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Ankur Joshi, Saket Kale, Satish Chandel, and D Kumar Pal. 2015. Likert scale: Explored and explained. *British journal of applied science & technology*, 7(4):396–403.

Antonios Kalloniatis and Panagiotis Adamidis. 2024. Computational humor recognition: a systematic literature review. *Artificial Intelligence Review*, 58(2):43.

Justine T Kao, Roger Levy, and Noah D Goodman. 2016. A computational model of linguistic humor in puns. *Cognitive science*, 40(5):1270–1285.

Dana Lahat, Tülay Adali, and Christian Jutten. 2015. Multimodal data fusion: an overview of methods, challenges, and prospects. *Proceedings of the IEEE*, 103(9):1449–1477.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Weixin Liang, Yuhui Zhang, Hancheng Cao, Binglu Wang, Daisy Yi Ding, Xinyu Yang, Kailas Vodrahalli, Siyu He, Daniel Scott Smith, Yian Yin, et al. 2024. Can large language models provide useful feedback on research papers? a large-scale empirical analysis. *NEJM AI*, 1(8):AIoa2400196.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55:1 – 35.

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):1–35.

Teresa Lucas. 2004. *Deciphering the meaning of puns in learning English as a second language: A study of triadic interaction*. Ph.D. thesis, The Florida State University.

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

Mishaim Malik, Muhammad Kamran Malik, Khawar Mehmood, and Imran Makhdoom. 2021. Automatic speech recognition: a survey. *Multimedia Tools and Applications*, 80:9411–9457.

Ruli Manurung, Graeme Ritchie, Helen Pain, Annalu Waller, Dave O'Mara, and Rolf Black. 2008. The construction of a pun generator for language skills development. *Applied Artificial Intelligence*, 22(9):841–869.

Justin McKay. 2002. Generation of idiom-based witticisms to aid second language learning. *Stock et al*, pages 77–87.

Tomas Mikolov. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 3781.

Tristan Miller. 2016. Adjusting sense representations for word sense disambiguation and automatic pun interpretation.

Tristan Miller, Christian F 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)*, pages 58–68.

Tristan Miller and Mladen Turković. 2016. Towards the automatic detection and identification of english puns. *The European Journal of Humour Research*, 4(1):59–75.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40.

Anirudh Mittal, Yufei Tian, and Nanyun Peng. 2022. Ambipun: Generating puns with ambiguous context. In *Association for Computational Linguistics (ACL)*.

Lili Mou, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2015. Backward and forward language modeling for constrained sentence generation. *arXiv: Computation and Language*.

Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, Andrew Y Ng, et al. 2011. Multimodal deep learning. In *ICML*, volume 11, pages 689–696.

Anton Nijholt, Andreea Niculescu, Alessandro Valitutti, and Rafael E Banchs. 2017. Humor in human-computer interaction: a short survey. In *16th IFIP TC13 International Conference on Human–Computer Interaction, INTERACT 2017*, pages 192–214. Indian Institute of Technology Madras.

Xuefei Ning, Zinan Lin, Zixuan Zhou, Zifu Wang, Huazhong Yang, and Yu Wang. 2023. Skeleton-of-thought: Large language models can do parallel decoding. *Proceedings ENLSP-III*.

Theo X Olausson, Alex Gu, Benjamin Lipkin, Cedegao E Zhang, Armando Solar-Lezama, Joshua B Tenenbaum, and Roger Levy. 2023. Linc: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. *arXiv preprint arXiv:2310.15164*.

OpenAI. 2023a. Gpt-3.5-turbo. https://platform.openai.com/docs/models/gpt-3-5-turbo. Accessed: 2025-01-05.

OpenAI. 2023b. Gpt-4 and gpt-4 turbo. https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo. Accessed: 2025-01-05.

Marinela Parović, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2022. Bad-x: Bilingual adapters improve zero-shot cross-lingual transfer. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1791–1799.

Adam Pauls and Dan Klein. 2012. Large-scale syntactic language modeling with treelets. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 959–968.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*.

Alec Radford. 2018. Improving language understanding by generative pre-training.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Sunny Rai and Shampa Chakraverty. 2020. A survey on computational metaphor processing. *ACM Computing Surveys (CSUR)*, 53(2):1–37.

C Ramakristanaiah, P Namratha, Rajendra Kumar Ganiya, and Midde Ranjit Reddy. 2021. A survey on humor detection methods in communications. In *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pages 668–674. IEEE.

Oleksandr Rebrii, Inna Rebrii, and Olha Pieshkova. 2022. When words and images play together in a multimodal pun: From creation to translation. *Lublin Studies in Modern Languages and Literature*, 46(2):85–97.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*.

Graeme Ritchie. 2005. Computational mechanisms for pun generation. In *Proceedings of the Tenth European Workshop on Natural Language Generation (ENLG-05)*.

Brian Roark, Lawrence Wolf-Sonkin, Christo Kirov, Sabrina J Mielke, Cibu Johny, Isin Demirsahin, and Keith Hall. 2020. Processing south asian languages written in the latin script: the dakshina dataset. *arXiv preprint arXiv:2007.01176*.

Yash Raj Sarrof. 2025. Homophonic pun generation in code mixed hindi english. In *Proceedings of the 1st Workshop on Computational Humor (CHum)*, pages 23–31.

Rustam Shadiev, Wu-Yuin Hwang, Nian-Shing Chen, and Yueh-Min Huang. 2014. Review of speech-to-text recognition technology for enhancing learning. *Journal of Educational Technology & Society*, 17(4):65–84.

Sakib Shahriar. 2022. Gan computers generate arts? a survey on visual arts, music, and literary text generation using generative adversarial network. *Displays*, 73:102237.

Qing Chen Shao, Zhen Zhen Wang, and Zhi Jie Hao. 2013. Contrastive studies of pun in figures of speech. *Advanced Materials Research*, 756:4721–4727.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2018. Ordered neurons: Integrating tree structures into recurrent neural networks. *arXiv preprint arXiv:1810.09536*.

KaShun Shum, Shizhe Diao, and Tong Zhang. 2023. Automatic prompt augmentation and selection with chain-of-thought from labeled data. *arXiv preprint arXiv:2302.12822*.

Robert E Smith, Jiemiao Chen, and Xiaojing Yang. 2008. The impact of advertising creativity on the hierarchy of effects. *Journal of advertising*, 37(4):47–62.

Włodzimierz Sobkowiak. 1991. *Metaphonology of English paronomasic puns*. Lang.

BE Stein. 1993. *The Merging of the Senses*. MIT Press.

Jiao Sun, Anjali Narayan-Chen, Shereen Oraby, Alessandra Cervone, Tagyoung Chung, Jing Huang, Yang Liu, and Nanyun Peng. 2022a. Expunations: Augmenting puns with keywords and explanations. *arXiv preprint arXiv:2210.13513*.

Jiao Sun, Anjali Narayan-Chen, Shereen Oraby, Shuyang Gao, Tagyoung Chung, Jing Huang, Yang Liu, and Nanyun Peng. 2022b. Context-situated pun generation. *arXiv preprint arXiv:2210.13522*.

Jiashuo Sun, Yi Luo, Yeyun Gong, Chen Lin, Yelong Shen, Jian Guo, and Nan Duan. 2023. Enhancing chain-of-thoughts prompting with iterative bootstrapping in large language models. *arXiv preprint arXiv:2304.11657*.

I Sutskever. 2014. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*.

Masahiro Suzuki and Yutaka Matsuo. 2022. A survey of multimodal deep generative models. *Advanced Robotics*, 36(5-6):261–278.

Yufei Tian, Divyanshu Sheth, and Nanyun Peng. 2022. A unified framework for pun generation with humor principles. *arXiv preprint arXiv:2210.13055*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

Bradley Tyler, Katherine Wilsdon, and Paul M Bodily. 2020. Computational humor: Automated pun generation. In *ICCC*, pages 181–184.

Alessandro Valitutti, Oliviero Stock, and Carlo Strapparava. 2009. Graphlaugh: a tool for the interactive generation of humorous puns. In *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, pages 1–2. IEEE.

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)*, pages 243–248.

Margot Van Mulken, Renske Van Enschot-van Dijk, and Hans Hoeken. 2005. Puns, relevance and appreciation in advertisements. *Journal of pragmatics*, 37(5):707–721.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Christopher Venour. 2000. *The computational generation of a class of pun.* Queen's University.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023. Openchat: Advancing open-source language models with mixed-quality data. *ArXiv*, abs/2309.11235.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zhijun Xu, Siyu Yuan, Lingjie Chen, and Deqing Yang. 2024. " a good pun is its own reword": Can large language models understand puns? *arXiv preprint arXiv:2404.13599*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. 2015. Humor recognition and humor anchor extraction. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2367–2376.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Zhiyuan Zeng, Xiaonan Li, Tianxiang Sun, Cheng Chang, Qinyuan Cheng, Ding Wang, Xiaofeng Mou, et al. 2024. Aggregation of reasoning: A hierarchical framework for enhancing answer selection in large language models. *arXiv preprint arXiv:2405.12939*.

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

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

Jingjie Zeng, Liang Yang, Jiahao Kang, Yufeng Diao, Zhihao Yang, and Hongfei Lin. 2024. "barking up the right tree", a gan-based pun generation model through semantic pruning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2119–2131.

Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, and In So Kweon. 2023a. Text-to-image diffusion models in generative ai: A survey. *arXiv preprint arXiv:2303.07909*.

Chenshuang Zhang, Chaoning Zhang, Sheng Zheng, Mengchun Zhang, Maryam Qamar, Sung-Ho Bae, and In So Kweon. 2023b. A survey on audio diffusion models: Text to speech synthesis and enhancement in generative ai. *arXiv preprint arXiv:2303.13336*.

Tuo Zhang, Tiantian Feng, Yibin Ni, Mengqin Cao, Ruying Liu, Katharine Butler, Yanjun Weng, Mi Zhang, Shrikanth S Narayanan, and Salman Avestimehr. 2024. Creating a lens of chinese culture: A multimodal dataset for chinese pun rebus art understanding. *arXiv preprint arXiv:2406.10318*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Wei Zhao, Steffen Eger, Johannes Bjerva, and Isabelle Augenstein. 2020. Inducing language-agnostic multilingual representations. *arXiv preprint arXiv:2008.09112*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Haotong Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *ArXiv*, abs/2306.05685.

Wei Zheng and Xiaolu Wang. 2023. Humor experience facilitates ongoing cognitive tasks: Evidence from pun comprehension. *Frontiers in Psychology*, 14:1127275.

Yukun Zhu. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *arXiv preprint arXiv:1506.06724*.

A Pun Categories

We outline the characteristics of different types of puns for clearer differentiation, including phonetic, graphic, meaning, and example, as shown in Table 3. "Same", "similar" and "different" respectively indicate whether the pun word and its substitute word same, similar, or different in phonic, graphic and meaning.

B Additional Evaluation

In this section, we supplemented additional details regarding the evaluation metrics.

Suprisal. Based on (He et al., 2019), the pun word w^p is more surprising relative to its alternative word w^a in the local context, while is less in the global context. Therefore, S_{ratio} is defined as a ratio to balance the metric:

$$S_{ratio} \begin{cases} -1, & S_{local} < 0 \text{ or } S_{global} < 0, \\ S_{local}/S_{global}, & \text{otherwise.} \end{cases} \tag{8}$$

where S_{local} and S_{local} are local surprisal and global surprisal, respectively. A higher value of S_{ratio} indicates a better-quality pun.

Dist-1 & Dist-2. We supplemented the formulation of Dist-2 here:

$$\text{Dist-2} = \frac{unique\ bigrams}{total\ generated\ bigrams} \tag{9}$$

where a higher Dist-2 score indicates greater diversity in the generated sentences, whereas a lower score suggests more generic and repetitive text.

C Dataset

The pun dataset for different types are summarized in Table 4. We list the datasets in five dimensions:

- The type of puns.
- The source of the datasets.
- The total number of the datasets.
- The language of the datastes.
- Is the dataset publicly available?

D Paper Collection

This section outlines the approach that we used to collect relevant papers in this survey. We initially searched for the keywords "pun research", "computational humour", and "pun dataset" on arXiv and Google Scholar, identifying a total of around 150 publications. Then, we filtered papers that specifically focused on pun generation, resulting in approximately 30 papers. Subsequently, we applied

Type	Phonic	Graphic	Meaning	Example
Homophonic Puns	Similar	Different	Different	Dentists don't like a hard day at the orifice (office).
Heterographic Puns	Same	Different	Different	Life is a puzzle, look here for the missing peace (piece).
Homographic Puns	Same	Same	Different	Always trust a glue salesman. They tend to stick to their word.
Visual Puns	N/A	N/A	Different	

Table 3: List of pun categories. N/A indicates that the element is not applicable.

Dataset	**Type**	**Source**	**Corpus (C)**	**Language**	**Availability**
Paron(Sobkowiak, 1991)	heg	Advertisements	3,850	English	✓
Paron-edit(Jaech et al., 2016)	heg	(Sobkowiak, 1991)	1,182	English	×
Church(Bell et al., 2011)	hog	Church	373	English	×
Pun-Yang(Yang et al., 2015)	N/A	Website	2,423	English	✓
Pun-Kao(Kao et al., 2016)	hop	Website	435	English	✓
Puns (Jaech et al., 2016)	N/A	Website	75	English	×
SemEval (Miller et al., 2017)	hog&heg	Experts	2,878	English	✓
SemEval-P (Miller et al., 2017)	hog	Experts	1,607	English	✓
SemEval-G (Miller et al., 2017)	heg	Experts	1,271	English	✓
ExPUNations (Sun et al., 2022a)	hog&heg	(Miller et al., 2017)	1,999	English	✓
CUP (Sun et al., 2022b)	hog&heg	(Miller et al., 2017)	2,396	English	✓
ChinesePun (Chen et al., 2024)	hop&hog	Website	2,106	Chinese	✓
ChinesePun-P (Chen et al., 2024)	hop	Website	1,049	Chinese	✓
ChinesePun-G (Chen et al., 2024)	hog	Website	1,057	Chinese	✓
Pun Rebus Art (Zhang et al., 2024)	visual	Museum	1,011	Multi-language	✓
UNPIE (Chung et al., 2024)	hog&heg	(Miller et al., 2017)	1,000	Multi-language	✓
UNPIE-P (Chung et al., 2024)	hog	(Miller et al., 2017)	500	Multi-language	✓
UNPIE-G (Chung et al., 2024)	heg	(Miller et al., 2017)	500	Multi-language	✓

Table 4: List of pun datasets. Hog, hop, heg and visual denote the types of homographic puns, homophonic puns, heterographic puns and visual puns, respectively. N/A indicates that the elements are not mentioned in the original paper.

System	**Type**	**Task**	**Language**
JAPE (Binsted and Ritchie, 1994)	heg & hog	Question-Answer	English
HCPP (Venour, 2000)	hop	Text Generation	English
WISCRAIC (McKay, 2002)	heg	Text Generation	English
PUNDA (Dybala et al., 2008)	heg & hog	Dialogue	Japanese
STANDUP (Manurung et al., 2008)	hop	Dialogue	English
T-PEG (Hong and Ong, 2009)	hop & hog	Text Generation	English
PAUL BOT (Tyler et al., 2020)	hop & hog	Dialogue	English
AliGator (Dehouck and Delaborde, 2025)	hop	Text Generation	French

Table 5: System of pun generation using traditional methods. Hog, hop and heg denote the types of homographic puns, homophonic puns and heterographic puns, respectively.

the forward and backward snowballing technique by examining the references and citations of these seed papers to identify additional relevant studies. We carefully reviewed all identified papers and ultimately compiled the findings into this survey.

E Traditional Systems

In this section, we summarized the pun generation systems with traditional methods in Section 4.1, as shown in table 5. We here listed the types of puns, task scenarios and languages corresponding to the system's applications.

F Related Surveys

To our knowledge, there are currently only surveys on computational humour research, while no focusing exclusively on puns. Amin and Burghardt (2020) provides a survey on humour generation, including generation systems, evaluation methods, and datasets. However, it does not specifically analyze the category of puns and only summarizes papers published prior to 2020. Nijholt et al. (2017) concluded a survey on designing humour and interacting with social media, virtual agents, social robots and smart environments. In addition, other humour studies have been examined from the perspectives of detection (Ramakrishtanaiah et al., 2021; Ganganwar et al., 2024) and recognition (Kalloniatis and Adamidis, 2024). Furthermore, there are some relevant surveys in creating writing such as poetry composition (Bena and Kalita, 2020; Elzohbi and Zhao, 2023), storytelling (Gieseke et al., 2021; Alhussain and Azmi, 2021), arts (Shahriar, 2022) and metaphor (Rai and Chakraverty, 2020; Ge et al., 2023). our survey provides a comprehensive overview of various methods focused on pun generation, including those published in recent few years.