TwistList: Resources and Baselines for Tongue Twister Generation

Tyler Loakman[1]*, **Chen Tang**[2]* **and Chenghua Lin**[1]†
[1]Department of Computer Science, The University of Sheffield, UK
[2]Department of Computer Science, The University of Surrey, UK
{tcloakman1,c.lin}@sheffield.ac.uk
chen.tang@surrey.ac.uk

Abstract

Previous work in phonetically-grounded language generation has mainly focused on domains such as lyrics and poetry. In this paper, we present work on the generation of tongue twisters - a form of language that is required to be phonetically conditioned to maximise sound overlap, whilst maintaining semantic consistency with an input topic, and still being grammatically correct. We present **TwistList**, a large annotated dataset of tongue twisters, consisting of 2.1K+ human-authored examples. We additionally present several benchmark systems (referred to as **TwisterMisters**) for the proposed task of tongue twister generation, including models that both do and do not require training on in-domain data. We present the results of automatic and human evaluation to demonstrate the performance of existing mainstream pre-trained models in this task with limited (or no) task specific training and data, and no explicit phonetic knowledge. We find that the task of tongue twister generation is challenging for models under these conditions, yet some models are still capable of generating acceptable examples of this language type.

1 Introduction

Phonetically constrained language generation is a primary subarea of computational creativity in natural language generation (NLG), primarily encompassing lyric and poetry generation (Tian and Peng, 2022; Wöckener et al., 2021; Xue et al., 2021; Zhang et al., 2020a; Agarwal and Kann, 2020), as well as pun generation (Sun et al., 2022; He et al., 2019; Yu et al., 2018), and continues to prove challenging for myriad reasons. Primarily, such works require the inclusion of phonetic factors such as metre and rhyme, which involves careful consideration of candidate vocabulary on the syllable level, leading to a reduced pool of allowable vocabulary once these constraints are in place.

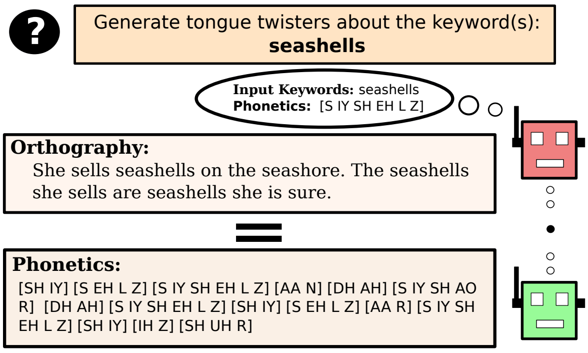

Generate tongue twisters about the keyword(s): **seashells**
Input Keywords: seashells
Phonetics: [S IY SH EH L Z]
Orthography:
She sells seashells on the seashore. The seashells she sells are seashells she is sure.
Phonetics:
[SH IY] [S EH L Z] [S IY SH EH L Z] [AA N] [DH AH] [S IY SH AO R] [DH AH] [S IY SH EH L Z] [SH IY] [S EH L Z] [AA R] [S IY SH EH L Z] [SH IY] [IH Z] [SH UH R]

Figure 1: Tongue Twister Generation aims to generate an utterance with high levels of phonetic overlap, requiring understanding of semantics, grammar, and phonetics.

In this paper, we present work on the generation of *tongue twisters*, a type of phonetically constrained language that is rarely explored in the NLG community. As a form of creative generation, tongue twisters can facilitate numerous useful applications, including: (1) being used as a pedagogical tool (Sugiharto et al., 2022; Somoff, 2014; Wilshire, 1999); (2) as a source of humorous entertainment stemming from unintentional mispronunciations; (3) as a stylistic device for engaging children in reading (e.g. Dr. Seuss stories (Geisel, 1965)); (4) as a method of designing memorable slogans and tag lines (Guerini et al., 2015); and (5) as stimuli in neuroscience/physiology research (Wong et al., 2019; O'Halloran, 2020; Kember et al., 2017).

Tongue twister generation posits unique challenges compared to other generation tasks. One of the most pertinent features of tongue twisters is the presence of high levels of phonetic overlap across tokens (Wilshire, 1999). Consequently, whilst other types of creative generation may require only *some* output tokens to consider phonetics (such as rhyme or syllable counts), tongue twisters present an extreme version of this problem where the phonetics of almost all generated tokens must be considered. This leads to a very small vocabulary from which to choose

[*] Equal contribution.
[†] Corresponding author.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 579–589
July 9-14, 2023 ©2023 Association for Computational Linguistics

semantically relevant words, and presents further challenges with maintaining grammatical validity.

The only work that we are aware of on tongue twister generation at the time of conducting this research is by Keh et al. (2022), who present models that train on graphemes and phonemes, and take either a starting prompt to be continued, or keywords around which to theme an output. They release *TT-Corp*, a dataset of 644 tongue twisters with parallel non-twister equivalents. We differentiate our work through the release of a dataset that is over 3x larger and which has undergone substantial human quality control. Furthermore, we assess the results of a wider range of popular pre-trained models on this task, including ChatGPT, without explicit injection of phonetic knowledge due to the difficulty in encoding phonetics and the expertise required to utilise phonetic characteristics appropriately. Our experimental results show that most popular pretrained language models (PLMs) rely on pure word repetition to generate tongue twisters, whilst some (i.e. BART) are able to generate more sophisticated examples. Additionally, very large zero-shot models (i.e. ChatGPT) are able to generate convincing tongue twisters almost on-par with human equivalents.[1]

To summarise our contributions, we present:

- **TwistList**, a large annotated dataset of human-authored tongue twisters, containing 2.1K+ examples with human evaluation of their quality.
- **TwisterMisters**, a series of baseline models for tongue twister generation using the most popular state-of-the-art PLMs.
- Extensive automatic and human evaluation to assess the ability of PLMs to implicitly model the complex phonetic phenomena in tongue twisters.

2 Related Works

Previous work in phonetically constrained generation has taken one of two approaches: 1) train a generation model on a collection of in-domain texts, or 2) train a generation model on prosaic out-of-domain text, with constraints imposed at decoding time. For example, Lau et al. (2018) collect 3,355 sonnets to produce novel poetry and train models to generate text in iambic pentameter, whilst Xue et al. (2021) train a rap generation model on 272,839 in-domain examples, infusing knowledge of rhythm afterwards. On the other hand, Van de Cruys (2020) train on a subset of CommonCrawl, imposing constraints on topic and rhyme as *a priori* distributions, whilst Tian and Peng (2022) train a title-to-keyword module on narrative texts in addition to a sonnet generation model trained on news articles and short stories from Reddit. They imposed literary techniques (simile/metaphor) and metre/rhyme constraints at decoding time, owing to the lack of sufficient training data.[2]

Dataset	Train	Val	Test	Total
# Tongue Twisters	1912	106	107	2128
Vocabulary Size	9556	946	880	10358
# Total Phonemes	55	43	46	56
# RAKE Keywords	3333	316	288	3567
# BERTopic Keywords	250	132	160	250
Avg. # Input Keywords (RAKE)	3.16	3.32	3.01	3.16
Avg. # Input Phonemes	5.57	5.83	5.16	5.56
Avg. Tongue Twister Length (Words)	15.01	16.59	13.54	15.01
Avg. # Input Phonemes	26.06	28.25	23.50	26.04

Table 1: The Statistics of **TwistList**.

3 Tongue Twister Generation

3.1 Task Definition

We formulate the task of tongue twister generation as follows: for a given set of keywords, we aim to generate a tongue twister T, whereby T comprises a sequence of words $\{w_1, w_2, ...w_n\}$. The generated output must satisfy the following constraints: (1) the output should be semantically related to the input keywords; (2) the output should show maximal levels of phonetic overlap across tokens; and (3) the output should be grammatically valid (Wilshire, 1999). Of these requirements, phonetic overlap is the most central to defining text as a "tongue twister".

3.2 TwistList Dataset

Dataset Construction. We present **TwistList**, an annotated dataset of 2.1K+ human-authored tongue twisters for use by the community. The examples contained therein come from a variety of sources available on the web.[3] For each tongue twister, phonetic transcription is provided using the *g2p-en* package,[4] in addition to keywords extracted with RAKE and BERTopic to represent the topic of the tongue twister. Following experimentation with both RAKE and BERTopic, only RAKE keywords are used in training due to human preference and issues regarding the use of BERTopic on short texts (where

[1] Our code and resources can be accessed at https://github.com/tangg555/TwistList

[2] Additionally, there is often a reluctance in computational creativity to train on examples, under the assumption that the newly generated content will be overly derivative.

[3] The source of each tongue twister is stated for each entry.

[4] https://pypi.org/project/g2p-en/

frequently no keywords are extracted). The main statistics of the dataset are presented in Table 1.

RAKE:	sells thick socks
BERTopic:	short shorts socks sock
Twister: **Phonetics:**	Seth at Sainsbury's sells thick socks. [S EH1 TH] [AE1 T] [S EY1 N S B ER0 IY0 Z] [S EH1 L Z] [TH IH1 K] [S AA1 K S]

Table 2: Example from TwistList

Quality Control. Quality control on our dataset was performed in multiple ways. Firstly, it was ensured that only sufficiently unique tongue twisters were kept in the dataset, as determined by removing examples with over 90% word overlap (rather than keeping variants of the same tongue twister, such as "Peter Piper picked a pickled pepper" versus "Peter *the* Piper picked..."). Additionally, non-standard spellings were manually converted to standard US English[5] to avoid G2P issues.[6] Similarly, tongue-twisters containing obscure vocabulary (such as medicine and dinosaur names) were excluded to further minimise errors. An annotation platform was developed (see Appendix A.1), with which 3 human evaluators, who are native speakers of English, were provided with 100 sampled instances from the dataset to rate the quality of the resulting tongue twisters and the associated extracted keywords. The full dataset contains 2,500+ tongue twisters, of which 2,128 are kept for training/development/testing after filtering examples with insufficient extracted keywords and excessive similarity to existing entries.

To summarise, 3 annotators evaluated the quality of the dataset, where 88% of assessed tongue twisters were considered high quality, and 6% considered "suitable" (Kappa = 0.321). An example from **TwistList** is provided in Table 2. As Table 4 shows, the final dataset can be considered high quality, owing to fair/moderate levels of approval and agreement across evaluators. Demographic information of the evaluators can be found in Appendix A.2.

3.3 Baseline Models

We present the following baseline models (dubbed **TwisterMisters**) for the task of tongue twister generation on our TwistList dataset:

Finetuned Baselines. For the finetuned baselines, we chose popular models for language generation, including **GPT-2** (Radford et al., 2019), **DialoGPT** (Zhang et al., 2020c), **T5** (Raffel et al., 2020), and **BART** (Lewis et al., 2020). These were finetuned with RAKE keywords extracted from human-authored tongue twisters as the input and the tongue twister text from **TwistList** as the target. This was in order to represent our baselines training on in-domain data. At inference time, the prompt "`Generate tongue twisters about the keyword(s): X`" is used, where X refers to the input consisting of one or more RAKE keywords extracted from tongue twisters. The full training details are given in Appendix A.3. We also conducted experiments on all aforementioned baselines without finetuning (i.e., a zero-shot setting), and the results were very poor. Therefore, we did not include these results in the paper.

Training-Free Baseline We additionally provide a TwisterMister baseline that does not require any training. We utilise OpenAI's **ChatGPT**[7] with the same prompt as a zero-shot setting for generation.[8] Each request to ChatGPT was submitted as part of a separate session, to avoid the effects of extended dialogue influencing outputs. ChatGPT has been utilised in order to set a practical upper-bound of what may be expected from models without explicit phonetic knowledge, owing to its wealth of training data and 175B parameter architecture.[9] It is assumed that ChatGPT's training data contains tongue twisters, and therefore it is able to abstract away the general patterns of such language in order to provide novel examples (though most likely based on graphemes rather than phonemes).

4 Experiments

Automatic Evaluation. We present the results of automatic evaluation on generated outputs and golden examples in Table 3 for the following metrics: **Perplexity** (**PPL**), **BLEU** (**B-1/B-2**) (Papineni et al., 2002), **ROUGE** (**R-1/R-2/R-L**) (Lin, 2004), and **BERTScore** Precision, Recall, and F-Measure (Zhang

[5] For example, where phonetic spellings or letter substitutions such as "k" for "c" were used for literary and visual effect, such as "kwik" for "quick".

[6] *g2p-en* uses the CMU Pronouncing Dictionary to retrieve transcriptions, which is an American English resource.

[7] `https://chat.openai.com/chat`

[8] No direct comparison is made to PANCETTA (Keh et al., 2022) as no code has been publicly released at the time of writing, and essential implementation details are absent from the paper.

[9] ChatGPT based on GPT-3.5, rather than GPT-4.

Model	PPL↓	B-1↑	B-2↑	R-1↑	R-2↑	R-L↑	PO↓	Init-PO↓	BS-P↑	BS-R↑	BS-F↑
GPT-2	8.40	0.007	0.003	1.301	0.123	1.315	0.022	0.020	0.690	0.810	0.744
DialoGPT	3.83	0.038	0.025	7.724	3.610	7.640	0.069	0.089	0.754	0.831	0.790
T5	10.16	0.057	0.038	9.701	4.573	9.574	0.689	0.727	0.795	0.818	0.806
BART	1.65	0.073	0.051	11.883	6.109	10.353	0.075	0.120	0.795	0.845	0.819
ChatGPT	N/A	0.200	0.137	36.765	20.659	33.437	0.093	0.157	0.888	0.894	0.883

Table 3: Results of Automatic Evaluation. Golden PO = 0.385 and Golden Init-PO = 0.417. Due to the one-to-many issue in creative language generation, we acknowledge that the referenced metrics are imperfect.

Choices (%)	Sample Quality			
	High.	Suitable.	Bad.	Kappa
RAKE keywords	**82.0**	18.0	0.0	0.321
BERTopic keywords	15.0	**85.0**	0.0	0.445
Tongue Twisters	**88.0**	6.0	4.0	0.321

Table 4: Kappa refers to Fleiss' Kappa (Fleiss, 1971). All results achieve fair or moderate agreement. Good tongue twisters that are deemed a bit longer (3%) or shorter (3%) than expected are considered "suitable".

et al., 2020b) (**BS-P/BS-R/BS-F**). PPL, BLEU and ROUGE are standard metrics in language generation to assess quality, whilst BERTScore assesses semantic similarity to a gold reference. Additionally, we propose two new metrics, Phonetic Overlap (**PO**) and Initial Phonetic Overlap (**Init-PO**). **PO** refers to the average overlap of all phonemes across tokens (# unique phonemes / # total phonemes), whereas **Init-PO** is the ratio of unique word-initial phonemes to the number of words (# unique word-initial phonemes / # words).

These phonetic metrics reward longer outputs. We argue that, all things equal, a longer tongue twister is better than a shorter one as it provides more entertainment and more opportunities for mispronunciation. Perfect scores on PO and Init-PO can be achieved by repetition of a single word. Whilst this does not lead to high quality outputs, these metrics are intended exclusively to be indicators of the phonetics, rather than an overall guide to quality. In both cases, higher levels of overlap results in lower ("better") scores, and the highest ("worst") achievable score is 1.

The results in Table 3 show rather clear scaling, with the performance ranking on most metrics (except Perplexity and phoneme overlap) being identical. On the models explicitly finetuned for this task, GPT-2 is shown to be the worst, whilst BART performs the best. We hypothesise that GPT-2's poor performance is in part due to its simple causal language modelling objective alongside its decoder-only architecture (which is also in DialoGPT). Furthermore, whilst T5 performed well on the automatic metrics, manual inspection revealed that T5 often misinterpreted the task from the prompt, choosing to select its own keywords from the entire prompt, rather than using only the provided keyword list. On the other hand, the training-free zero-shot model, ChatGPT, was shown to perform best on all metrics. This is to be expected as ChatGPT has over 50x more parameters than any other tested PLM, with various pre-training objectives and reinforcement learning, leading to performant zero-shot capabilities. This further demonstrates that PLMs struggle to learn phonetic patterns implicitly from text, especially in English, which has high levels of irregular orthography. Furthermore, with limited data, PLMs struggle to learn the unusual probability distributions underlying tongue twisters, where word choices are intentionally "twisted", obscure, and anti-euphonious. Additionally, due to the wealth of training data seen by ChatGPT, it is likely that many examples have been seen during training.

Human Evaluation. Due to tongue twisters being a creative domain where articulation abilities are tested, we also perform human evaluation. 3 evaluators were asked to rate 100 outputs from the best performing standard baseline (BART), in addition to ChatGPT outputs and gold examples from **TwistList** on the following criteria: **Relevance** (how relevant the tongue twister is given the keyword inputs), **Fluency** (how grammatically valid the output is), **Difficulty of Articulation** (how difficult a tongue twister is to say), **Cohesion** (how much sense the output makes), and **Entertainment Value** (how entertaining the output is, considering sounds and semantics). All ratings were on a 5-point Likert scale. Evaluator demographics and training materials are in Appendix A.2.

The mean scores of human evaluation (Table 5) fall in line with expectations, with *golden* examples performing best on all metrics, and ChatGPT placing second on all but Difficulty of Articulation.[10] BART is able to produce outputs that are deemed to be the

[10] We exclude relevance for the golden examples as these were collected from the web, not elicited with keyword prompts.

Score (1 to 5)	Human Evaluation		
	BART	ChatGPT	Golden
Relevance	4.667*	4.971†	N/A
Difficulty of Articulation	4.143*	4.102*	**4.291***
Fluency	3.028**	4.915**	**4.938****
Coherence	3.217*	4.798*	**4.909***
Entertainment Value	3.269*	4.070*	**4.254***

Table 5: Results of Human Evaluation. The best scores are in **bold**, and the second best are underlined. We calculate Fleiss' Kappa for each metric, and mark the agreement fair*, moderate** and substantial†.

second most difficult to articulate, which we infer may be the result of slight morphological variants of input keywords being used repeatedly, making distinguishing between them during articulation quite challenging (whilst not being able to exploit deeper phonetic relations). The moderate score on Fluency (3.028) suggests instances of poor grammar may also hinder articulation abilities when expected grammatical structures are not found, leading to an interaction between grammatical validity and articulatory difficulty. Additionally, ChatGPT scoring the lowest for articulatory difficulty may be due to occasionally misunderstanding the requirements of a tongue twister, sometimes producing rhymes or standard prose (see Appendix A.4). However, ChatGPT scores well for Relevance and Fluency, highlighting its capability in producing high-quality coherent language. Perhaps most interestingly, none of the BART score averages on any human evaluation criteria fall below 3 ("neither agree nor disagree"). This performance is therefore quite good for a model finetuned on only 2128 examples, with no additional phonetic knowledge.

Input	assistant assist
GPT-2	assistant assist assistant assist assistant
DialogGPT	assistant assistant assistant assistant assistant assistant assistant assistant
T5	assistant assist assistant
BART	A assistant assist is an assistant assist, assistants assist to assist assistants.
ChatGPT	Assistant ants assist ants in carrying leaves to the ant hill.
Golden	If I assist a sister-assistant, will the sister's sister-assistant assist me?

Table 6: Example outputs for the input "assistant assist". "Golden" refers to the human-authored tongue twisters.

5 Case Study

Within the example in Table 6, GPT-2 resorts to simply repeating the input, successfully achieving phonetic overlap, but failing to be grammatically valid or particularly sophisticated. This pattern is also demonstrated by DialoGPT and T5. Conversely, BART is able to introduce tokens unseen in the input to create an almost grammatically valid output (the primary mistake being indefinite article agreement, where in the first instance "an" would have been correct, rather than "a"). BART's output is also semantically and logically coherent, with "A assistant assist is an assistant assist" being valid (yet redundant), and "assistants assist to assist assistants" also being comprehensible. This example demonstrates why evaluators with high English proficiency and language/linguistics education were selected, as the same word may have different parts of speech, creating outputs that seem grammatically invalid, but do actually follow the rules of English.[11] Further investigation is needed to ascertain whether the models are intentionally exploiting this lexical ambiguity, or if human evaluators are demonstrating apophenia, where patterns are found in what is effectively noise (Brugger, 2001). Finally, ChatGPT utilises morphology to exploit the similarity of the plural noun "assistants" and the phrase "assist ants", and provides a continuation that is in line with the expected behaviour of ants. In comparison to the golden example, ChatGPT's output may be considered more interesting topic-wise, at the expense of not being as phonetically complex ("carrying leaves to the ant hill" contributes heavily to semantics, whilst not being recognisable as part of a tongue twister). For further analysis, please see Appendix A.4.

6 Conclusion

We present work on the topic of tongue twister generation, a form of phonetically-constrained language generation that aims to maximise phonetic overlap, whilst conveying meaningful semantics. We motivate the potential application domains for such generated language, and provide a large annotated dataset of tongue twisters, **TwistList**, to encourage further work. Finally, we present a series of benchmark models alongside automatic/human evaluation to assess generation quality.

[11] https://en.wikipedia.org/wiki/Buffalo_buffalo_Buffalo_buffalo_buffalo_buffalo_Buffalo_buffalo

Limitations

Whilst the system presented within this paper is capable of allowing human-in-the-loop contributions (via selecting the input keywords on which to condition the output), it is not able to produce tongue-twisters that take advantage of particular features of speech sounds such as place and manner of articulation, in order to create more advanced outputs that exploit phonetic relatedness (rather than exact matches). The same can be said of our proposed metrics, PO and Init-PO, which do not account for phonetic similarity across sounds that share manner/place of articulation (e.g. "**sh**e **s**ells **s**ea **sh**ells"). Additionally, whilst commonly known tongue twisters may follow a particular format (e.g. rhyme schemes), such schemes and templates have not been enforced here. We also do not demonstrate the capabilities of these systems if they were trained on phonetic transcriptions explicitly, as we only aim to assess their performance when training on graphemes in standard orthography.

Ethics Statement

All use of human participants in this study has been approved by the Ethics Board of the primary author's institution, including the disclosure of demographic information. Regarding the generation of tongue twisters, language generation is a necessarily creative domain that has the ability to reproduce content that some individuals may find offensive. Care was taken to check outputs in the human evaluation set for any such materials, and if they had been produced, they would have been removed from the evaluation set. Additionally, no egregiously offensive material has been provided in the TwistList dataset. However, the distinction between offensive and humorous content is a highly complex topic, and therefore some examples within the dataset may not be suitable for all individuals (e.g. suggestive content and swearing, such as "I'm not the pheasant plucker, I'm the pheasant plucker's son", and the clear relation to common expletives).

Acknowledgements

Tyler Loakman is supported by the Centre for Doctoral Training in Speech and Language Technologies (SLT) and their Applications funded by UK Research and Innovation [grant number EP/S023062/1]. Chen Tang is supported by the China Scholarship Council (CSC) for his doctoral study (File No.202006120039).

References

Rajat Agarwal and Katharina Kann. 2020. Acrostic poem generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1230–1240, Online. Association for Computational Linguistics.

Peter Brugger. 2001. From haunted brain to haunted science: A cognitive neuroscience view of paranormal and pseudoscientific thought. In James Hournan and RenseEditors Lange, editors, *Hauntings and Poltergeists: Multidisciplinary Perspectives*, page 195–213. McFarland.

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

Theodore Seuss Geisel. 1965. *Fox in socks: Dr. Seuss's book of tongue tanglers*. Random House.

Marco Guerini, Gözde Özbal, and Carlo Strapparava. 2015. Echoes of persuasion: The effect of euphony in persuasive communication. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1483–1493, Denver, Colorado. Association for Computational Linguistics.

He He, Nanyun Peng, and Percy Liang. 2019. Pun generation with surprise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1734–1744, Minneapolis, Minnesota. Association for Computational Linguistics.

Henglin Huang, Chen Tang, Tyler Loakman, Frank Guerin, and Chenghua Lin. 2022. Improving Chinese story generation via awareness of syntactic dependencies and semantics. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*.

Sedrick Scott Keh, Steven Y. Feng, Varun Gangal, Malihe Alikhani, and Eduard Hovy. 2022. Pancetta: Phoneme aware neural completion to elicit tongue twisters automatically.

Heather Kember, Kathryn Connaghan, and Rupal Patel. 2017. Inducing speech errors in dysarthria using tongue twisters. *International journal of language & communication disorders*, 52(4):469–478.

Jey Han Lau, Trevor Cohn, Timothy Baldwin, Julian Brooke, and Adam Hammond. 2018. Deep-speare: A joint neural model of poetic language, meter and rhyme. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1948–1958, Melbourne, Australia. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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

Ken D. O'Halloran. 2020. A tongue-twister to translation? increased complexity of genioglossus movement during wakefulness in persons with obstructive sleep apnoea. *The Journal of Physiology*, 598(3):435–436.

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

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.

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.

Victoria Somoff. 2014. Four is not fourteen: Tongue twister patterns and the unmastery of language. *Western Folklore*, 73(2/3):195–215.

Prasetyawan Sugiharto, Yan Santoso, and Maila Shofyana. 2022. Teaching english pronunciation using tongue twister. *Acitya: Journal of Teaching and Education*, 4(1):189–197.

Jiao Sun, Anjali Narayan-Chen, Shereen Oraby, Shuyang Gao, Tagyoung Chung, Jing Huang, Yang Liu, and Nanyun Peng. 2022. Context-situated pun generation. In *EMNLP 2022*.

Chen Tang, Chenghua Lin, Henglin Huang, Frank Guerin, and Zhihao Zhang. 2022a. EtriCA: Event-triggered context-aware story generation augmented by cross attention. In *Findings of the Association for Computational Linguistics: EMNLP 2022*.

Chen Tang, Hongbo Zhang, Tyler Loakman, Chenghua Lin, and Frank Guerin. 2022b. Terminology-aware medical dialogue generation. *arXiv preprint arXiv:2210.15551*.

Chen Tang, Zhihao Zhang, Tyler Loakman, Chenghua Lin, and Frank Guerin. 2022c. NGEP: A graph-based event planning framework for story generation. In *Proceedings of AACL-IJCNLP*, Online.

Yufei Tian and Nanyun Peng. 2022. Zero-shot sonnet generation with discourse-level planning and aesthetics features. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3587–3597, Seattle, United States. Association for Computational Linguistics.

Tim Van de Cruys. 2020. Automatic poetry generation from prosaic text. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2471–2480, Online. Association for Computational Linguistics.

Carolyn E. Wilshire. 1999. The "tongue twister" paradigm as a technique for studying phonological encoding. *Language and Speech*, 42(1):57–82.

Jörg Wöckener, Thomas Haider, Tristan Miller, The-Khang Nguyen, Thanh Tung Linh Nguyen, Minh Vu Pham, Jonas Belouadi, and Steffen Eger. 2021. End-to-end style-conditioned poetry generation: What does it take to learn from examples alone? In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 57–66, Punta Cana, Dominican Republic (online). Association for Computational Linguistics.

Min Ney Wong, Yanky Chan, Manwa L. Ng, and Frank F. Zhu. 2019. Effects of transcranial direct current stimulation over the broca's area on tongue twister production. *International Journal of Speech-Language Pathology*, 21(2):182–188. PMID: 29642741.

Lanqing Xue, Kaitao Song, Duocai Wu, Xu Tan, Nevin L. Zhang, Tao Qin, Wei-Qiang Zhang, and Tie-Yan Liu. 2021. DeepRapper: Neural rap generation with rhyme and rhythm modeling. 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)*, pages 69–81, Online. Association for Computational Linguistics.

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

Rongsheng Zhang, Xiaoxi Mao, Le Li, Lin Jiang, Lin Chen, Zhiwei Hu, Yadong Xi, Changjie Fan, and Minlie Huang. 2020a. Youling: an AI-assisted lyrics creation system. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 85–91, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020c. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

A Appendices

A.1 Dataset Quality Control

An annotation platform was developed as shown in (Figure 2).

A.2 Human Participants

Due to tongue twisters being highly reliant on articulation abilities, the demographics of the human participants used within this work are highly important. Additionally, tongue twisters are also a form of humour and entertainment, where individual perceptions of what may or may not be considered humorous or entertaining differ according to numerous factors. In an effort to remain as transparent as possible, and follow best practices for human evaluation, relevant demographic information of participants are outlined below (with the necessary requisite permission and ethical approval).

Dataset Evaluation All evaluators involved in the quality control process of the **TwistList** dataset are native speakers of English, and either have or are working towards University level qualifications. Additionally, 2 of the 3 evaluators have extensive education in linguistics or modern languages. No monetary incentive was provided.

Generation Evaluation All evaluators involved in the evaluation of the quality of generated tongue twisters are native speakers of English, and either hold or are working towards University level qualifications in Linguistics, Modern Languages or NLP. Additionally, all evaluators cited the United Kingdom as their country of socialisation, and no participants reported language processing difficulties that could affect results. No monetary incentive was provided.

Materials Provided to Human Participants Additionally, all evaluators for both the dataset and generation outputs were presented with calibration examples to demonstrate the sort of outputs that would be presented, and the logic behind particular scores, in order to minimise individual interpretations of the scoring criteria. All evaluation was performed on a custom made online annotation platform (Figure 3).

A.3 Training Details

All pre-trained models used (naturally excluding ChatGPT) are based on publicly available checkpoints from Hugging Face.[12] Models are trained for up to 5 epochs on a Tesla A5000 machine with the best checkpoints selected based on the validation loss. The batch size is set to 32, and the learning rate is $8e^{-5}$, with the Adam optimiser selected for training. To help the loss curve converge on our small few-shot dataset, we limit the generation length to 100 (covering all test tongue twisters). Meanwhile, the source length is limited to 150. The training and testing steps are set up with the implementation of the PyTorch Lightning[13] framework to guarantee the reliability of the experiment. All language models are fairly trained and tested with the same steps.

A.4 Further Qualitative Comments

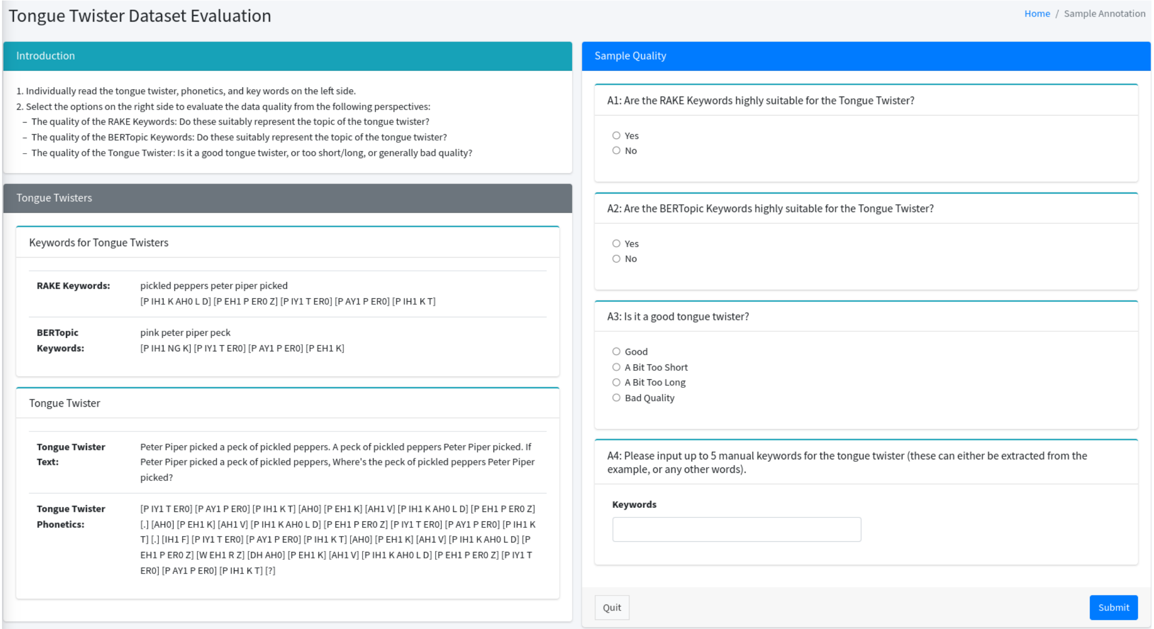
Whilst the pattern of extreme word repetition is seen in many of the finetuned models (often with the exception of BART, which is demonstrated to be capable of producing slightly more sophisticated outputs), overall assessment of the tongue twisters produced at inference time reveals interesting patterns, particularly in regard to ChatGPT outputs. Firstly, the limits of ChatGPT are made apparent in a few examples such as the input "silver shiny ship sank" generating "How much wood would a woodchuck chuck if a woodchuck could chuck silver shiny ships?", a clear derivation of a famous woodchuck related tongue twister that it is rather safe to assume appears multiple times in ChatGPTs training material. Additionally, comments from evaluators also reveal that ChatGPT's output is often considered more of a rhyme or general literary text, rather than specifically a tongue twister. However, examples such as these are also found in the human-authored golden examples, demonstrating that there is no steadfast consistent opinion as to what constitutes a (good) tongue twister. Likewise, some examples may contain large amounts of sound repetition, but not in a way that necessarily presents articulatory difficulty.

A.5 Future Works

In this paper, we mainly analyse the performance of large-scale pretrained language models (PLMs) on Tongue Twister Generation, and propose a corresponding dataset for further investigation. In further works, we aim to propose novel models which can better leverage phonetic symbols. There

[12] https://huggingface.co/models

[13] https://www.pytorchlightning.ai/



Tongue Twister Dataset Evaluation

Introduction

1. Individually read the tongue twister, phonetics, and key words on the left side.
2. Select the options on the right side to evaluate the data quality from the following perspectives:
  - The quality of the RAKE Keywords: Do these suitably represent the topic of the tongue twister?
  - The quality of the BERTopic Keywords: Do these suitably represent the topic of the tongue twister?
  - The quality of the Tongue Twister: Is it a good tongue twister, or too short/long, or generally bad quality?

Tongue Twisters

Keywords for Tongue Twisters

RAKE Keywords:	pickled peppers peter piper picked [P IH1 K AH0 L D] [P EH1 P ER0 Z] [P IY1 T ER0] [P AY1 P ER0] [P IH1 K T]
BERTopic Keywords:	pink peter piper peck [P IH1 NG K] [P IY1 T ER0] [P AY1 P ER0] [P EH1 K]

Tongue Twister

Tongue Twister Text:	Peter Piper picked a peck of pickled peppers. A peck of pickled peppers Peter Piper picked. If Peter Piper picked a peck of pickled peppers, Where's the peck of pickled peppers Peter Piper picked?
Tongue Twister Phonetics:	[P IY1 T ER0] [P AY1 P ER0] [P IH1 K T] [AH0] [P EH1 K] [AH1 V] [P IH1 K AH0 L D] [P EH1 P ER0 Z] [.] [AH0] [P EH1 K] [AH1 V] [P IH1 K AH0 L D] [P EH1 P ER0 Z] [P IY1 T ER0] [P AY1 P ER0] [P IH1 K T] [.] [IH1 F] [P IY1 T ER0] [P AY1 P ER0] [P IH1 K T] [AH0] [P EH1 K] [AH1 V] [P IH1 K AH0 L D] [P EH1 P ER0 Z] [W EH1 R Z] [DH AH0] [P EH1 K] [AH1 V] [P IH1 K AH0 L D] [P EH1 P ER0 Z] [P IY1 T ER0] [P AY1 P ER0] [P IH1 K T] [?]

Sample Quality

A1: Are the RAKE Keywords highly suitable for the Tongue Twister?

○ Yes
○ No

A2: Are the BERTopic Keywords highly suitable for the Tongue Twister?

○ Yes
○ No

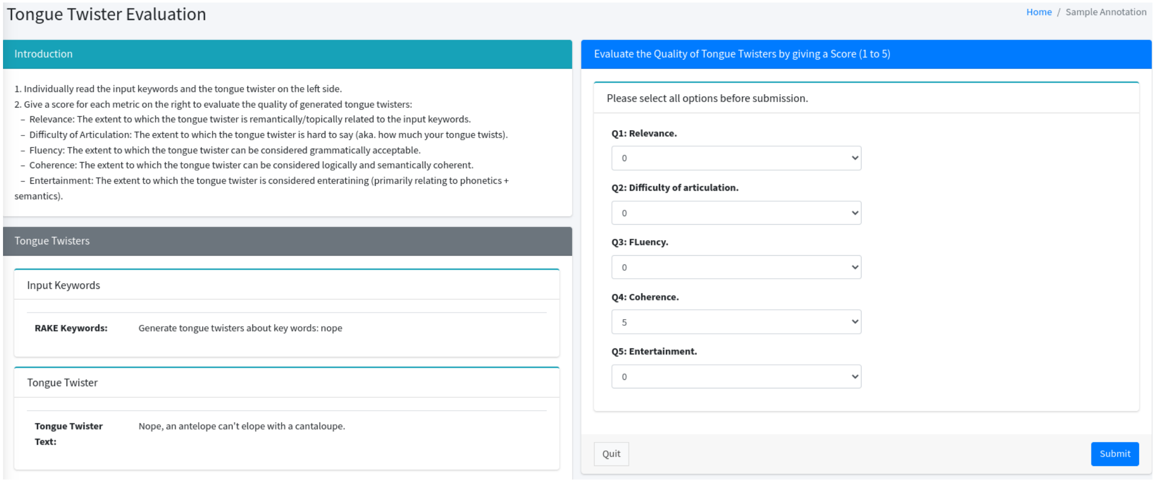
A3: Is it a good tongue twister?

○ Good
○ A Bit Too Short
○ A Bit Too Long
○ Bad Quality

A4: Please input up to 5 manual keywords for the tongue twister (these can either be extracted from the example, or any other words).

Keywords

Figure 2: TwistList Quality Control Annotation Platform



Tongue Twister Evaluation

Introduction

1. Individually read the input keywords and the tongue twister on the left side.
2. Give a score for each metric on the right to evaluate the quality of generated tongue twisters:
  - Relevance: The extent to which the tongue twister is remantically/topically related to the input keywords.
  - Difficulty of Articulation: The extent to which the tongue twister is hard to say (aka. how much your tongue twists).
  - Fluency: The extent to which the tongue twister can be considered grammatically acceptable.
  - Coherence: The extent to which the tongue twister can be considered logically and semantically coherent.
  - Entertainment: The extent to which the tongue twister is considered enteratining (primarily relating to phonetics + semantics).

Tongue Twisters

Input Keywords

RAKE Keywords:	Generate tongue twisters about key words: nope

Tongue Twister

Tongue Twister Text:	Nope, an antelope can't elope with a cantaloupe.

Evaluate the Quality of Tongue Twisters by giving a Score (1 to 5)

Please select all options before submission.

Q1: Relevance. 0

Q2: Difficulty of articulation. 0

Q3: FLuency. 0

Q4: Coherence. 5

Q5: Entertainment. 0

Figure 3: Human Evaluation Platform for Generated Outputs

are numerous existing works (Huang et al., 2022; Tang et al., 2022a,b) that provide approaches for injecting such knowledge into PLMs. However, the phonetic features differ from these text-format knowledge items, as phonemes are hard to encode with input text tokens when feeding into PLM encoders. Another promising approach is to explicitly model the phonetic features into text sequences (Tang et al., 2022c), though there is no observed method for transforming phonetic notation. We intend to perform further research based on these existing approaches.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Yes, in the required Limitations section as well as Section 4 (concerning our proposed metrics)

☑ A2. Did you discuss any potential risks of your work?
Ethics Statement

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract (all) and contribution summary at the end of the introduction.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

TwistList dataset (Section 3.2)

☑ B1. Did you cite the creators of artifacts you used?
Sources of all entries in the dataset are credited in the .json file for each entry.

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We did not discuss the licensing around our dataset. The dataset uses works that are freely available on the web and come from various sources such as websites, blogs, and ebooks. Many of these cases are Public Domain, and for those that are not, we believe we are in accordance with Fair Use, as the dataset does not encroach on the use case of the original works (no graphic design/other elements are maintained) and the dataset is for use as a research tool only. We will also reply promptly to any cases of copyright infringement that relevant copyright holders make us aware of.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
See answer to B2.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
See the Ethics Statement regarding the potential for tongue twisters to be offensive. Additionally, all tongue twisters are believed to be about fictional characters, rather than individuals.

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Such details are not explicitly stated. However, it can be easily ascertained from the paper that the tongue twisters we focus on are entirely in English (and the range of domains the tongue twisters were taken from can be seen in the "source" entry for each example).

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
See Table 1 for dataset statistics.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

Section 4 (page 3)

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix A.3

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix A.3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Tables 3/5. Scores are the mean, as is standard.

☒ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Exact details of evaluation implementations (except Phonetic Overlap) were not detailed. This is in part due to these metrics (BLEU/ROUGE/BERTScore) not being very reliable for creative language generation, and therefore the exact values from different implementations are not likely to be of use.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Section 3.2 and Section 4. In addition to Appendix A.2

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Screenshots of the annotation platforms can be found in Figures 2 and 3 in the Appendix

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
We declared that no monetary incentive was given to participants. We did not specify the recruitment process, but due to participants all holding or working towards university level qualifications, it can be inferred that they are colleagues.

☒ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
This information was not deemed necessary in the submitted paper (due to the limited risk of the data we were working with). However, it is stated in the Ethical Statement and Appendix A.2 that all shared information about human demographics was collected with the necessary permissions and approval.

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Ethical approval was gained for human evaluation of the dataset and generated outputs from the relevant institution

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
We provide demographic information for human participants in Appendix A.2