# ByGPT5: End-to-End Style-conditioned Poetry Generation with Token-free Language Models

**ACL 2023**

Jiaxin Zhang, Yashar Moshfeghi

University of Strathclyde

Part 1.

# Background

- **End-to-end fine-tuning of pre-trained language models**
  - GPT-2 or T5 on downstream tasks has been a popular training paradigm for text generation in the last few years
  - End-to-end models learn to complete a task by directly learning all steps
    - Without intermediary algorithms such as hand-crafted rules or post-processing
  - This approach has proven to be highly effective on a wide range of problems
    - Dialogue generation, Summarization, Machine translation
  - All these applications have in common
    - Only concern themselves with the generation of prosaic texts

Part 1.

# Background

- **Generating formal verse poetry remains a difficult problem**
  - Strict constraints on aesthetic style such as rhyme scheme, meter and alliteration

**BYGPT5 Generated quatrain with rhyme scheme, alliterations, and meter**

**Rhyme: 'ABAB' / Meter : "iambus(약강격)"**
**Alliteration: "medium"**

**When the sun shall set, and the moon be set,**
**And the night be thin and drear,**
**When the clouds break, and the great world be met**
**With the stars and moon so near,**

**Rhyme: 'ABBA' / Meter : "iambus"**
**Alliteration: "medium"**

**But I know the hour is near,**
**When the wind is in the east,**
**And the sun is on the feast,**
**And the moon is on the bier,**

**Rhyme: 'ABAB' / Meter : " trochee (강약격)"**
**Alliteration: "medium"**

**And the word that he spoke had been a sound of sorrow,**
**A sound that had stayed in the darkness without a sound,**
**Sound of swords and clangour of torment, call of to-morrow,**
**Sound of wars and thunder of battles and the blood of foes that won.**

Part 1.

# Background

- **Rhyme, Meter, Alliteration**
  - Strict constraints on aesthetic style such as rhyme scheme, meter and alliteration

**Rhyme & Meter**

**Hum**pty **Dum**pty **sat** on a **wall**,
**Hum**pty **Dum**pty **had** a great **fall**.
**All** the **king's** horses and **all** the king's **men**
**Cou**ldn't **put** Humpty **to**gether a**gain**.

**Alliteration**

**"The daily diary of the American dream."**
**(The Wall Street Journal Commercials)**

**"What we want is Watneys."**
**(Watney's Commercials)**

Part 1.

# Background

- **Rhyme, Meter, Alliteration**
  - **Rhyme**
    - Repetition of the same or similar sounds in the final accented syllables of words, which must be preceded by differing consonant
  - Imperfect rhymes
    - Final sounds are different or the words are identical
  - **Meter**
    - Rhythmic pattern within a verse
    - Accented-syllabic: Succession of stressed (—) and unstressed syllables (⏑) occurs at regular intervals



W. Harmon, C.H. Holman, and W.F. Thrall. A Handbook to Literature. Handbook to Literature Series. Prentice Hall. 2000.

Part 1.

# Background

- **Rhyme, Meter, Alliteration**
  - **Meter**
    - Rhythmic pattern within a verse
    - Accented-syllabic: Succession of stressed (—) and unstressed syllables (◡) occurs at regular intervals
  - The rhythmic unit: Foot
    - The meter of a verse can thus be described as a sequence of feet
    - Iambic (◡ —) , trochaic (— ◡) , anapestic (◡ ◡ —) , Dactylic (— ◡ ◡).

◡ — ◡ — ◡ — ◡ —
*The sweet wild strain, the sudden start,* A

◡ — ◡ — ◡ — ◡ —
*Which shakes the perfumed altar's flame,* B

◡ — ◡ — ◡ — ◡ —
*To make its shrine a sacred name,* B

◡ — ◡ — ◡ — ◡ —
*And sing its praise in every heart.* A

— ByGPT5



W. Harmon, C.H. Holman, and W.F. Thrall. A Handbook to Literature. Handbook to Literature Series. Prentice Hall. 2000.

Part 1.

# Background

- **Rhyme, Meter, Alliteration**
  - **Alliteration**
    - The repetition of the same consonant sounds or any vowel sounds at the beginning of words or syllables that are close together in a verse
    - Alliteration is secondary to rhyme and meter, follows less strict constraints
    - Therefore not as easily classified
  - In this work, we thus consider the level of alliteration instead, which we classify as either low, medium, or high

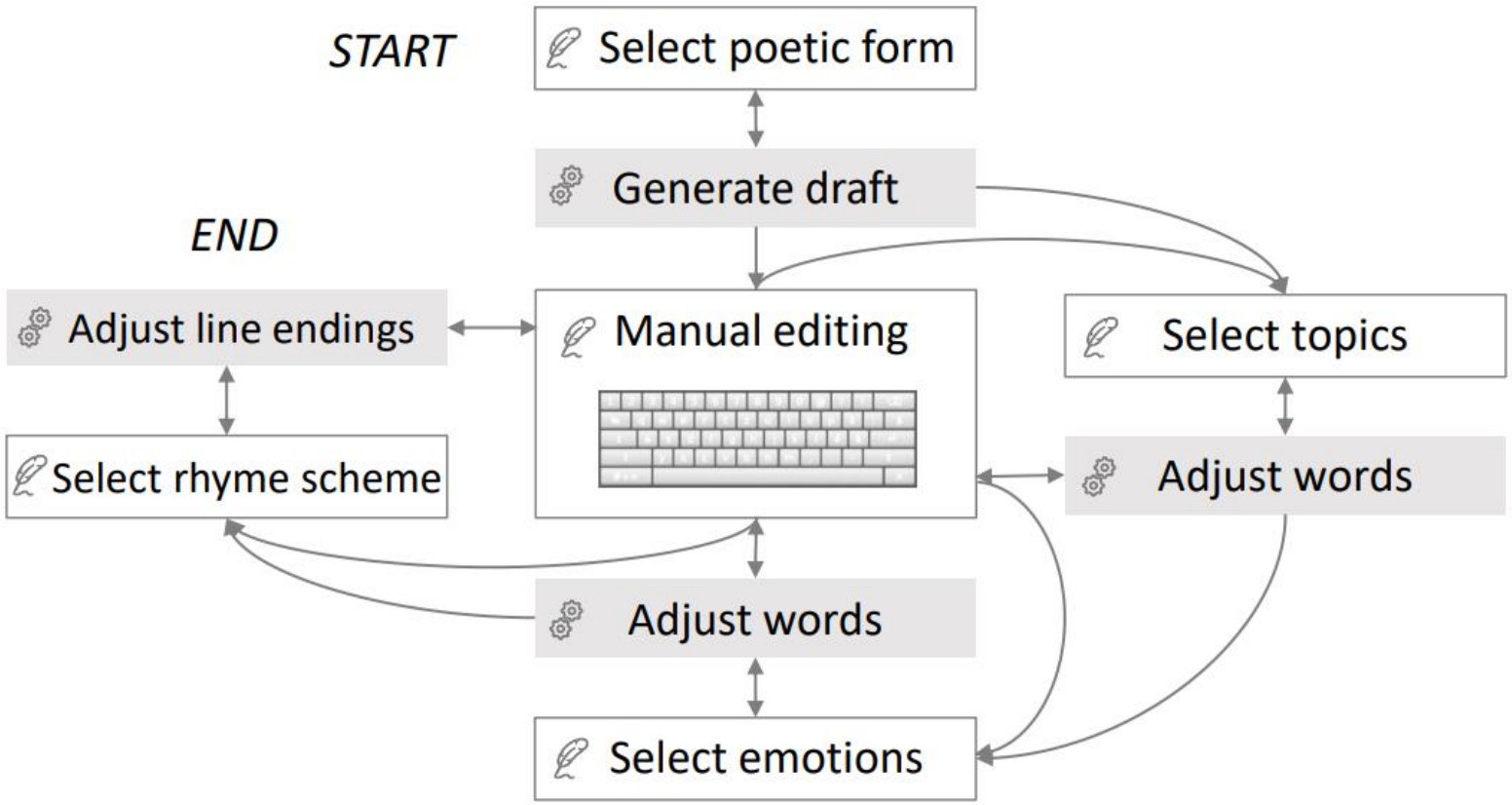


W. Harmon, C.H. Holman, and W.F. Thrall. A Handbook to Literature. Handbook to Literature Series. Prentice Hall. 2000.

Part 1.

# Background

- **Generating formal verse poetry remains a difficult problem**
  - Poetry generation systems rely on human guidance




**CR-PO System**

- TextGenRNN generate poem

| | Topic | Fluency |
|---|---|---|
| *Word selection* | | |
| TextGenRNN is better | 17.3 | 24.4 |
| IQ values are better | **67.7** | **41.7** |
| The two are similar | 14.9 | 33.9 |
| *Word replacement* | | |
| TextGenRNN is better | 14.3 | 3.1 |
| CamemBERT is better | **68.3** | **79.2** |
| The two are similar | 17.5 | 17.7 |

Table 3: Answers of human judges (%) for each method of word selection and replacement.



Andrei Popescu-Belis et al. Constrained Language Models for Interactive Poem Generation. LREC. 2022.

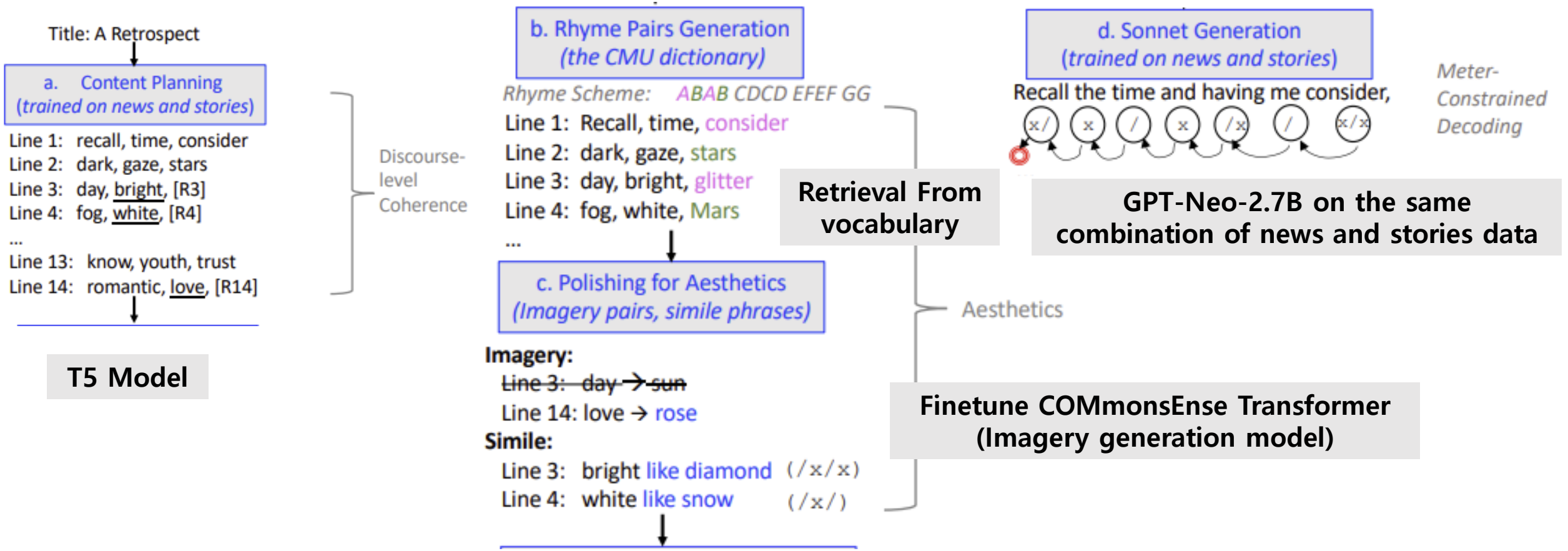
# Background

- **Generating formal verse poetry remains a difficult problem**
  - Injecting prior knowledge in the form of hard-coded constraints to filter model outputs or modify probability distributions
  - Break the whole process down into sophisticated task-specific model pipelines

Part 1.

# Background

- **Sonnet generation framework with four distinct pipeline steps**






Yufei Tian and Nanyun Peng. Zero-shot sonnet generation with discourse-level planning and aesthetics features. NAACL. 2022.

Part 1.

# Background

- **Cristina Garbacea and Qiaozhu Mei (2022)**
  - Incorporating prior knowledge generally becomes more difficult when heterogeneous constraints are involved or the number of constraints increases
  - Standard text-generation architectures do not lend themselves well for manually applying constraints
  - Due to the autoregressive generation of tokens from left to right, constraints at arbitrary positions cannot be implemented easily or only with additional trade-offs
  - For example, end rhymes, which come at the end of a verse, cannot be constrained in isolation due dependencies on previously generated tokens



Cristina Garbacea and Qiaozhu Mei. 2022. Why is constrained neural language generation particularly challenging?

# Introduction

- **Reduce the amount of human supervision in poetry generation and explore viable end-to-end solutions**
  - Hypothesize that failing to do so far has the following root causes:
  - (i) Lack of available training data
    - Poetry corpora labeled with aesthetic styles are few and rare
    - Speculate that they do not suffice to train a generalized model
  - (ii) Unfavorable tokenization algorithms
    - Aesthetic styles of poetry such as rhyme, meter, and alliteration are often expressed at the character-level
    - Otherwise, most available off-the-shelf pre-trained models operate at the subword-level

  ➡ Assumption:
    - Character-level models would perform similarly well at poetry generation

# Introduction

- **Contributions**
  - (i) Pre-train ByGPT5, to our knowledge the first decoder-only transformer for character-level language modeling
  - (ii) Create QuaTrain, a large machine-labeled poetry corpus of quatrains in German and English
  - (iii) Learn character-level styles better than subword-based systems, such as GPT-2 and mT5

    Token-free models (ByT5, ByGPT5), while being more parameter efficient and compared to humans

Linting Xue et al. ByT5: Towards a token-free future with pre-trained byte-to-byte models. Transactions of the Association for Computational Linguistics, 10:291–306. 2022.
Linting Xu et al. mT5: A massively multilingual pre-trained text-to-text transformer. ACL. 2021b.

Part 3.

# Character-Level Model (ByT5)

- **ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models**
  - One common choice is to assign a unique token ID to each word in a fixed finite vocabulary
  - An issue with using a fixed vocabulary of words
    - There is no obvious way to process a piece of text that contains an out-of-vocabulary word
    - E.g., map all unknown words to the same <UNK> token
    - This prevents the model from distinguishing between different out-of-vocabulary words

➡ A solution to the out-of-vocabulary problem: Subword tokenizers



Linting Xue et al. ByT5: Towards a token-free future with pre-trained byte-to-byte models. Transactions of the Association for Computational Linguistics, 10:291–306. 2022.

# Character-Level Model (ByT5)

- **Subword tokenizers**
  - Instead of mapping each word to a single token
  - Flexibility
    - Decompose words into smaller subword units with a goal of minimizing the total length of the token sequences for a fixed vocabulary size
  - Mispredictions
    - Typos, variants in spelling and capitalization, and morphological changes can all cause the token representation of a word or phrase to change completely

➡ Token-free NLP models that do not rely on a learned vocabulary to map words or subword units to tokens
  - Such models operate on raw text directly



Linting Xue et al. ByT5: Towards a token-free future with pre-trained byte-to-byte models. Transactions of the Association for Computational Linguistics, 10:291–306. 2022.

Part 3.

# Character-Level Model (ByT5)

- **Token-free models**
  - Text data is generally stored as a sequence of bytes
    - Feeding byte sequences directly into the model enables the processing of arbitrary text sequences
  - This approach is well-aligned with the philosophy of end-to-end learning
    - It endeavors to train models to directly map from raw data to predictions
  - A concrete benefit in terms of model size
    - **The large vocabularies of word- or subword-level models** often result in many parameters being devoted to the vocabulary matrix
    - **A byte-level model by definition** only requires 256 embeddings



Linting Xue et al. ByT5: Towards a token-free future with pre-trained byte-to-byte models. Transactions of the Association for Computational Linguistics, 10:291–306. 2022.

Part 3.

# Character-Level Model (ByT5)

- **Token-free models**
  - Migrating word representations out of a sparse vocabulary matrix and into dense network layers
    - Allow models to generalize more effectively across related terms (e.g., book / books) and orthographic variations
  - Models with a fixed vocabulary can complicate adaptation to new languages and new terminology
  - By definition, token-free models can process any text sequence



Linting Xue et al. ByT5: Towards a token-free future with pre-trained byte-to-byte models. Transactions of the Association for Computational Linguistics, 10:291–306. 2022.

Part 3.

# Character-Level Model (ByT5)

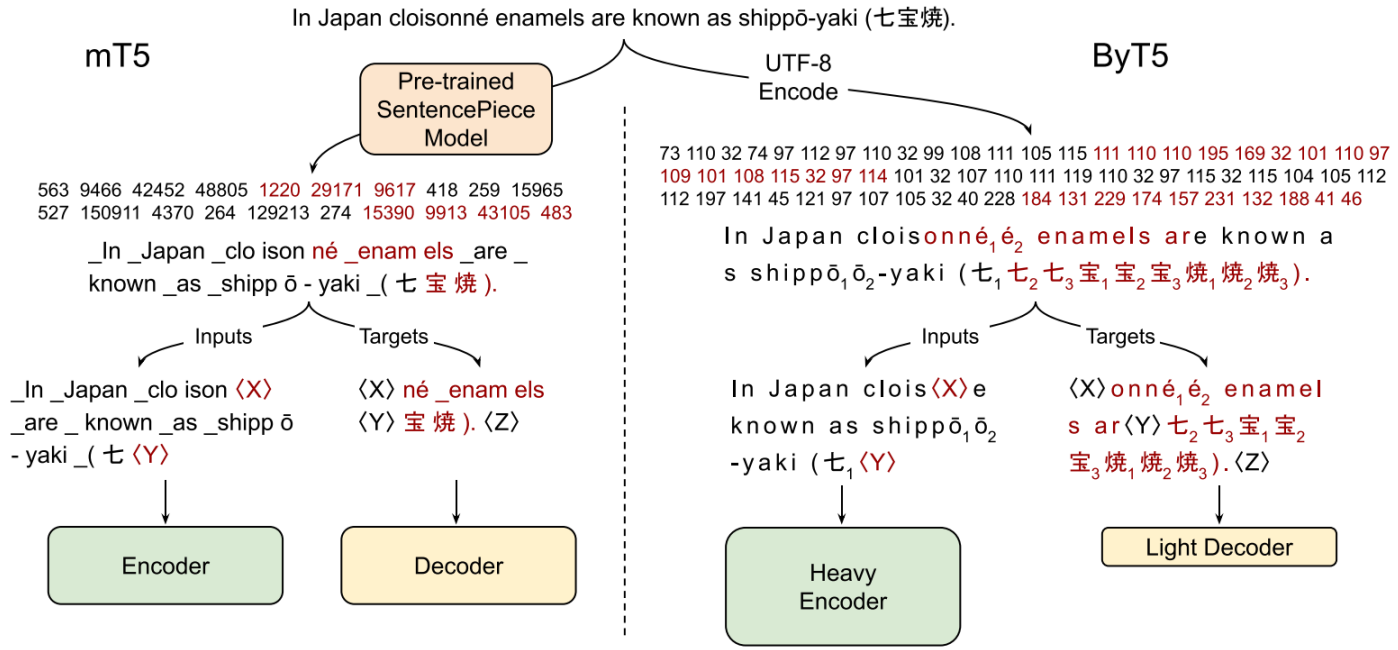
- **The main drawback of byte-level models**
  - Byte sequences tend to be significantly longer than token sequences
  - Computational costs of machine learning models tend to scale with sequence length
  - Much previous work on character- and byte-level models has explored ways to process long sequences efficiently using convolutions with pooling or adaptive computation time



Linting Xue et al. ByT5: Towards a token-free future with pre-trained byte-to-byte models. Transactions of the Association for Computational Linguistics, 10:291–306. 2022.

# Character-Level Model (ByT5)

- **Character-Level Model (ByT5)**
  - Without a dramatically unfavorable increase in computational cost
  - T5 framework : The design stays fairly close to mT5 (the multilingual variant of T5)
    - All text-based NLP problems are cast to a text-to-text format
    - Makes it simple to tackle an NLP task by generating a sequence of bytes






Linting Xue et al. ByT5: Towards a token-free future with pre-trained byte-to-byte models. Transactions of the Association for Computational Linguistics, 10:291–306. 2022.

# Character-Level Model (ByT5)

- **Changes from mT5: Dispense with the SentencePiece vocabulary**
  - Feed UTF-8 bytes directly into the model without any text preprocessing
  - The bytes are embedded to the model hidden size using a vocabulary of 256 possible byte values
  - An additional 3 IDs are reserved for special tokens: padding, end-of-sentence, and an unused <UNK> token that we include only for convention



Linting Xue et al. ByT5: Towards a token-free future with pre-trained byte-to-byte models. Transactions of the Association for Computational Linguistics, 10:291–306. 2022.

# Character-Level Model (ByT5)

- **Changes from mT5: Modify the pre-training task**
  - mT5 uses the “span corruption” pre-training objective
  - Rather than adding 100 new tokens for the sentinels, we find it sufficient to reuse the final 100 byte IDs
  - While mT5 uses an average span length of 3 subword tokens, we find that masking longer byte-spans is valuable
  - Specifically, we set our mean mask span length to 20 bytes, and show ablations of this value in Section 6



Linting Xue et al. ByT5: Towards a token-free future with pre-trained byte-to-byte models. Transactions of the Association for Computational Linguistics, 10:291–306. 2022.

# Character-Level Model (ByT5)

- **Changes from mT5: Decouple the depth of the encoder and decoder stacks**
  - Heavier encoder
    - Set our encoder depth to 3 times that of the decoder
    - Make the model more similar to encoder-only models like BERT
  - Light decoder
    - One might expect quality to deteriorate on tasks like summarization that require generation of fluent text
    - Heavy-encoder byte models performing better on both classification and generation tasks



Linting Xue et al. ByT5: Towards a token-free future with pre-trained byte-to-byte models. Transactions of the Association for Computational Linguistics, 10:291–306. 2022..

# Character-Level Model (ByT5)

- **Not all byte sequences are legal according to the UTF-8 standard**
  - Drop any illegal bytes in the model's output3 (though we never observed our models predicting illegal byte sequences in practice)
  - Like mT5, we set our sequence length to 1024 (bytes rather than tokens)
  - Train for 1 million steps over batches of 220 tokens



Linting Xue et al. ByT5: Towards a token-free future with pre-trained byte-to-byte models. Transactions of the Association for Computational Linguistics, 10:291–306. 2022.

Part 3.

# Character-Level Model (ByT5)

- **Not all byte sequences are legal according to the UTF-8 standard**
  - Drop any illegal bytes in the model's output3 (though we never observed our models predicting illegal byte sequences in practice)

**English Classification Tasks**

| | GLUE | | SuperGLUE | |
|---|---|---|---|---|
| Model | mT5 | ByT5 | mT5 | ByT5 |
| Small | 75.6 | **80.5** | 60.2 | **67.8** |
| Base | 83.0 | **85.3** | 72.5 | **74.0** |
| Large | **87.6** | 87.0 | **81.9** | 80.4 |
| XL | **88.7** | 87.9 | **84.7** | 83.2 |
| XXL | **90.7** | 90.1 | **89.2** | 88.6 |

**English Generation Tasks**

| | GEM-XSum (BLEU) | | TweetQA (BLEU-1) | | DROP (F1 / EM) | |
|---|---|---|---|---|---|---|
| Model | mT5 | ByT5 | mT5 | ByT5 | mT5 | ByT5 |
| Small | 6.9 | **9.1** | 54.4 | **65.7** | 40.0 / 38.4 | **66.6 / 65.1** |
| Base | 8.4 | **11.1** | 61.3 | **68.7** | 47.2 / 45.6 | **72.6 / 71.2** |
| Large | 10.1 | **11.5** | 67.9 | **70.0** | 58.7 / 57.3 | **74.4 / 73.0** |
| XL | 11.9 | **12.4** | 68.8 | **70.6** | 62.7 / 61.1 | **68.7 / 67.2** |
| XXL | 14.3 | **15.3** | 70.8 | **72.0** | 71.2 / 69.6 | **80.0 / 78.5** |



Linting Xue et al. ByT5: Towards a token-free future with pre-trained byte-to-byte models. Transactions of the Association for Computational Linguistics, 10:291–306. 2022.

# Models

- **End-to-end poetry generation systems for English and German by fine-tuning pretrained transformer models**
  - Two architectural variants:
    - Encoder-decoder transformers
    - Decoder-only transformers
  - Do not experiment with models with more than 400 million parameters
    - Exceed the capacity of our available GPU resources

# Models

- **End-to-end poetry generation systems for English and German by fine-tuning pretrained transformer models**
  - Encoder-Decoder
    - Encoder-decoder transformers
    - Decoder-only transformers
  - Decoder-only
    - As the input for encoder-decoder models is a relatively short sequence of styles, this could lead to an underutilization of the encoder
    - Hypothesis:

      Decoder-only model, with styles supplied as a prompt string, would be better suited for our task

      ➡ Refer to it as ByGPT5

# Models

- **ByGPT5**
  - Modifying the architecture of ByT5 and discard its encoder component
  - Initialize the weights with the decoder of ByT53 to warm-start the training process
  - Repeat this for the three smallest variants of ByT5
    - ByT5 has an asymmetrical architecture
    - The resulting models retain only 25% of its parameters
  - Training data
    - Openwebtext2 (Gao et al., 2021) for English
    - cc100 (Conneau et al., 2020) for German

Leo Gao et al. The pile: An 800gb dataset of diverse text for language modeling. 2021.
Alexis Conneau et al. Unsupervised cross-lingual representation learning at scale. ACL. 2020.

# Datasets

- **Quatrain**
  - Collect a range of labeled and unlabeled datasets of English and German poetry
  - Label them automatically: Real quatrains with pseudo-quatrains (any consecutive sequence)

| Dataset | Language | Verses | R | M | A |
|---|---|---|---|---|---|
| EPG64 | English | 1k | ✓ | ✓ | ✗ |
| PROSODIC | English | 2k | ✗ | ✓ | ✗ |
| FORB | English | 1k | ✓ | ✓ | ✗ |
| CHICAGO | English | 95k | ✓ | ✗ | ✗ |
| ANTI-K | German | 4k | ✓ | ✓ | ✗ |
| GRC | German | 41k | ✓ | ✗ | ✗ |
| EPG | English | 2.8m | ✗ | ✗ | ✗ |
| DLK | German | 2.8m | ✗ | ✗ | ✗ |
| QUATRAIN | English | 2.7m* | ✓ † | ✓ † | ✓ † |
| | German | 5.9m* | ✓ † | ✓ † | ✓ † |

* Verses may occur in multiple pseudo-quatrains.
† Labels are obtained from classifiers.

Part 5.

# Datasets

- **Automatically labeling rhyme and meter**
  - Leverage the available labeled data and train a range of classifiers
  - Held-out gold data and subsequently use the best performing classifier for each style
    - Unlabeled poetry: English Project Gutenberg (EPG), Deutsches Lyrik Korpus (DLK) (Haider, 2021)
    - Labeled poetry: Prosodic, Chicago Rhyme Corpus (Chicago), For-better-for-verse (FORB) (Tucker, 2011), German Rhyme Corpus (GRC) (Haider and Kuhn, 2018), as well as EPG64 and Anti-K (Haider, 2021)
    - Map meters which appear less than 25 times in our labeled corpora to the special label other

| Meter | Symbol |
|---|---|
| iambus | ⏑ – |
| trochee | – ⏑ |
| amphibrach | ⏑ – ⏑ |
| anapest | ⏑ ⏑ – |
| dactyl | – ⏑ ⏑ |
| alexandrine | ⏑ – ⏑ – ⏑ – ‖ ⏑ – ⏑ – ⏑ – |

# Datasets

- **Automatically labeling rhyme and meter**
  - Meter classification
    - A multiclass classification problem with a single verse as input
    - Classify the meter of a quatrain by choosing the dominant meter among the verses
  - Rhyme classification
    - A binary classification problem with two verses separated by a special token as input
    - The rhyme scheme by determining which verses rhyme and which do no

**F1-Score on classifying rhyme and meter**

| Model | Rhyme | Meter |
|---|---|---|
| CANINE-C | **98.05** | **58.49** |
| XLM-R | 97.22 | 54.65 |
| MBERT | 97.17 | 49.01 |

# Datasets

- **Labeling for alliteration**
  - The quantification of the level of alliteration in a document is a long known research problem (Skinner, 1939; Leavitt, 1976; Blain, 1987; Benner, 2014)
  - Let $vi$ be the atomic units of sound in verse $v$, Blain (1987) quantify alliteration
    f(·): a similarity function of two sounds; the default simply testing for equality
  - allit(·) counts alliterative sounds in a verse, applies a distance penalty, and normalizes the score to [0, 1]

$$\text{allit}(v) = \frac{\sum_{i=1}^{|v|} \sum_{j=i+1}^{|v|} \frac{\text{f}(v_i, v_j)}{j-i}}{\sum_{i=1}^{|v|} \sum_{j=i+1}^{|v|} \frac{1}{j-i}}$$

# Datasets

- **Labeling for alliteration**
  - Consider initial phonemes of words, as well as all further stressed phonemes as atomic sound units $vi$ , and to determine phonemes and stress
  - Use a graphemeto-phoneme conversion model (Zhu et al., 2022)
  - Conduct an internal study to determine several intensity thresholds based on a sample of quatrains
  - Classify the alliteration level of a quatrain as **low** if the score is below 0.05, **medium** if the score is below 0.1, and **high** if it is above that

# Experiment

- **Automatic Evaluation**
  - Rhyme schemes (AABB, ABAB, ABBA, and ABCB)
  - The most popular meters per language
    - iambus, trochee, anapest, and dactyl for English
    - iambus, trochee, and alexandrine for German
  - All levels of alliteration
  - Create 75 poems per model for each possible combination
  - To find out if styles are properly reflected in generated quatrains we reuse the classifiers

# Experiment

- **Automatic Evaluation**
  - Rhyme Score
    - Computes the recall of verses that should rhyme in a quatrain, as well as the recall of verses that should not and takes their arithmetic average
  - Alliteration Score
    - 1 if a quatrain has the correct alliteration level, else 0.
  - Meter Score
    - The fraction of verses with correctly classified meters
  - Coherence
    - Uses BERT for next sentence prediction to assess discourse relations of verses
    - The score is the fraction of consecutive verse pairs that are correctly classified to come after one another

# Experiment

- **Automatic Evaluation**

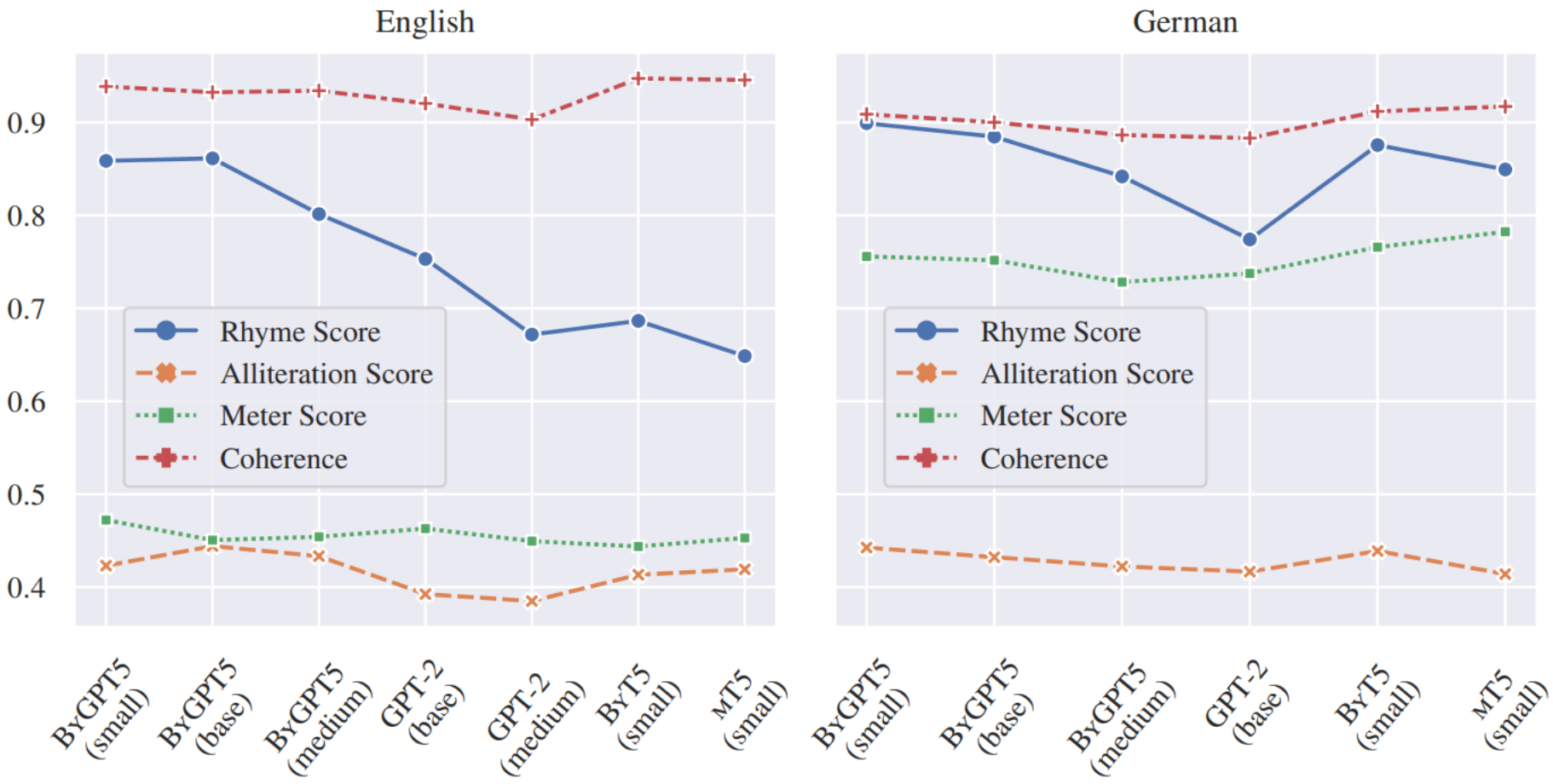



Figure 3: Automatic evaluation results for all models on English and German.

Part 6.

# Experiment

- **Human Evaluation**

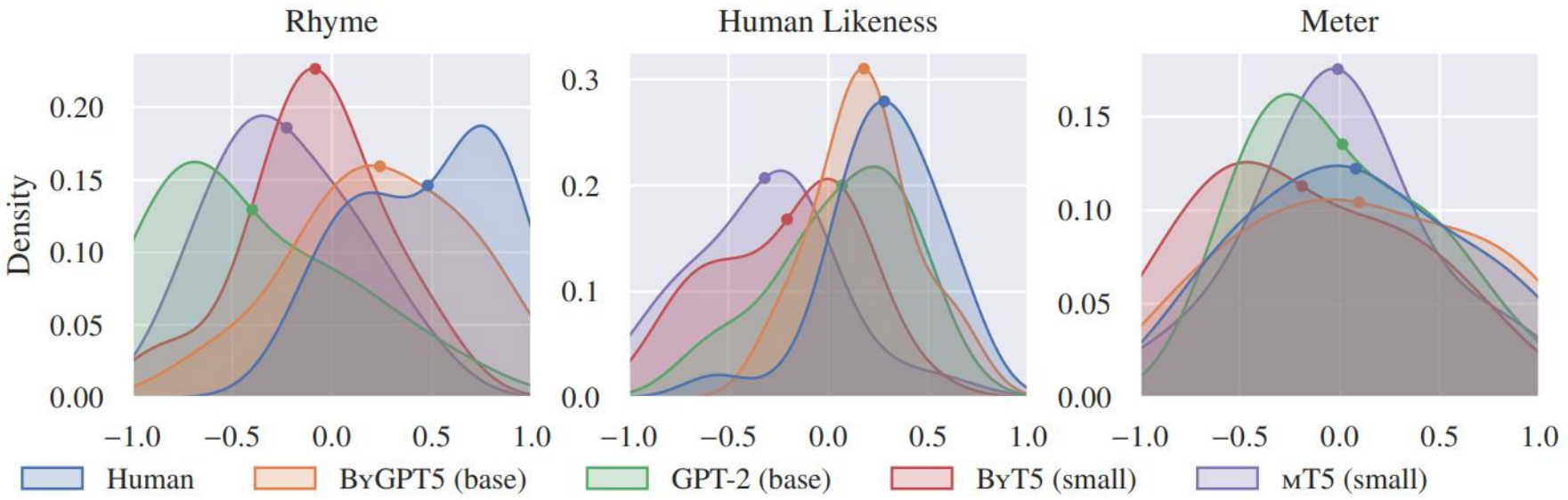



Figure 4: Distributions of BWS scores for rhyme, human likeness, and meter annotations through kernel density estimation. Scores range from -1 (very bad) to 1 (very good). The "•" markers denote expected values.

| Model | Rhyme | Human Likeness |
|---|---|---|
| Human | **0.84** | **0.89** |
| ByGPT5 | 0.57 | 0.55 |
| GPT-2 | 0.35 | 0.55 |
| ByT5 | 0.45 | 0.19 |
| mT5 | 0.28 | 0.32 |

Part 6.

# Conclusion

- **Automatic Evaluation**
  - End-to-end style conditioned poetry generation systems for quatrains in English and German
    - Generate poetry without the need for human supervision, except for the use of poetic training data
  - Token-free decoder-only language model, and show that fine-tuning it on a custom poetry corpus outperforms other models, such as GPT-2, mT5, and ByT5
    - Perform favorably against human poets in our constrained setting
  - Future Work
    - Extend our system to other poetic forms such as sonnets, limericks, or villanelles

# Appendix

- **Additional example poems generated with ByGPT5 (base) in German and English.**

| German | | English | |
|---|---|---|---|
| *Ein Reiter steht am Hafen,* | A | *With languid smile, the stealing tear retires,* | A |
| *Der schaut die Flut nicht an,* | B | *And the slow fading light on trembling fires!* | A |
| *Er hört die Schiffer schlafen* | A | *Now she receives the golden circlet round,* | B |
| *Im stillen Ozean.* | B | *And fills the woven chambers with a sound;* | B |
| *Schweigend stehn die Burgen nieder,* | A | *The first who learned the lesson there* | A |
| *Und die Lüfte sind verhallt,* | B | *Had learned to scoff and scorn to sneer,* | A |
| *Und die Trommeln klingen wider,* | A | *And that the learned might have been* | B |
| *Und die Büchsen knallen halt.* | B | *A shameless woman and a queen.* | B |

# Appendix

- **Additional example poems generated with ByGPT5 (base) in German and English.**

| German | | English | |
|---|---|---|---|
| *Der Greis erbebt, die Hand erstarrt,* | A | *Then came the labour of the day within,* | A |
| *Die Kinder schauern vor dem Sterben;* | B | *The gray beginning of the week,* | B |
| *Die Stimme bricht, die Thräne fallt,* | C | *And down we went with hope and terror sin,* | A |
| *Sie sieht ihm nach mit nassem Beben.* | B | *And not a word to say and speak.* | B |
| *Die Strömung wiederum durch alle Glieder dringt,* | A | *For the stars are in the sky;* | A |
| *Und alles, was da lebt und wallt und leuchtet, singt.* | A | *And the stars have gone to die* | A |
| *Da steigt ein Palmenstrauch aus dem erhabnen Lichte,* | B | *With their songs of joy and fear,* | B |
| *Er schwimmt auf einer Fluth und singet in Gedichte.* | B | *With their music and their cheer.* | B |