`Lingxi`: A Diversity-aware Chinese Modern Poetry Generation System

Xinran Zhang[1, 2, 3], **Maosong Sun**[1, 2, 3, 4*], **Jiafeng Liu**[1, 2, 3], **Xiaobing Li**[1, 2, 3]

[1]Department of AI Music and Music Information Technology, Central Conservatory of Music
[2]Key Laboratory of Music and Brain Science, Central Conservatory of Music, Ministry of Education, China
[3]Laboratory of AI Music, Central Conservatory of Music,
Laboratory of Philosophy and Social Sciences, Ministry of Education, China
[4]Department of Computer Science and Technology, Tsinghua University
zhangxr.wspn@gmail.com, sms@tsinghua.edu.cn

Abstract

Chinese modern poetry generation has been a challenging task. One issue is the Chinese word segmentation (CWS) which is critical to comprehend the Chinese language but was not always considered in common tokenization methods. Another is the decoding (sampling) method which may induce repetition and boredom and severely lower the diversity of the generated poetry. To address these issues, we present Lingxi, a diversity-aware Chinese modern poetry generation system. For the CWS issue, we propose a novel framework that incorporates CWS in the tokenization process. The proposed method can achieve a high vocabulary coverage rate with a reasonable vocabulary size. For the decoding method and the diversity issue, we propose a novel sampling algorithm that flattens the high likelihood part of the predicted distribution of the language model to emphasize the comparatively low-likelihood words and increase the diversity of generated poetry. Empirical results show that even when the top 60% of cumulative probability mass of the predicted distribution is flattened, our method achieves comparable or even better performance than baseline sampling methods. Our system is available at http://lingxi.website[†].

1 Introduction

Chinese modern poetry generation has been a challenging natural language processing (NLP) task. One issue is the *Chinese word segmentation* (CWS) problem. Unlike English words that are naturally delimited by white spaces, Chinese words do not have explicit word delimiters. Since different CWS strategies may completely change the semantics of Chinese words, CWS is always crucial for humans to comprehend the Chinese language and is regarded as a critical Chinese NLP task. Despite its importance, CWS is always ignored in benchmark tokenization methods such as byte pair encoding (BPE, Gage, 1994; Sennrich et al., 2016) or unigram language model (Unigram LM, Kudo, 2018), as well as in recent researches on Chinese poetry or lyric generation (e.g., systems by Lee et al., 2019, Zhang et al., 2020, and Zhang et al., 2022). A critical issue is that the rendered vocabulary from CWS tends to exhibit an extremely long "tail", which demands additional techniques to process before being incorporated into the language model.

Another challenge is the high diversity requirement for poetry generation. The task is unique compared with common neural generation tasks, which try to predict the correct answers from the training corpus and focus on the fluency of the generated texts. Concretely, a piece of easily understood and highly fluent poetry with high-likelihood words might not always be considered poetic, while semantically ambiguous poetry with low-likelihood words might be diversified and creative. We consider the diversity issue from the view of the decoding (sampling) algorithm of the language model. The widely acknowledged golden methods include the nucleus sampling (Holtzman et al., 2020), top-k sampling (Fan et al., 2018; Holtzman et al., 2018), and temperature sampling, which have been proved to be able to control the quality and diversity of generated texts. However, in our system, these golden methods might occasionally generate boring and repetitive samples with low diversity, severely hurting the quality of the generated poetry. Thus it inspires us to explore novel decoding algorithms to address the diversity issue for poetry generation.

To address these challenges, we present Lingxi, a diversity-aware Chinese modern poetry generation system with the following features. For the CWS issue, we propose a novel framework that incorporates CWS in the tokenization process. The proposed method not only leverages the human knowledge from the CWS model but also com-

[*]Corresponding author.

[†]Video demonstration is available at https://youtu.be/ofNTZFCM4DQ.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 3: System Demonstrations, pages 63–75
July 10-12, 2023 ©2023 Association for Computational Linguistics

bines the advantage of frequency-based tokenization methods, which can achieve a high coverage rate of the vocabulary while maintaining a reasonable vocabulary size. For the high-diversity issue, we propose a novel decoding (sampling) method, namely the *nucleus sampling with flattened head* (NS-FH) algorithm. It flattens the high-likelihood part of the predicted distribution to suppress the high-likelihood words and emphasize the less likely words to improve the diversity of the generated poetry. Surprisingly, we find that even if the top 60% of cumulative probability mass of the vocabulary's distribution is flattened, the model achieves comparable or even better performance than baseline methods.

2 The CWS Framework for Chinese Modern Poetry Corpus

2.1 The "Long-tail" Issue of CWS

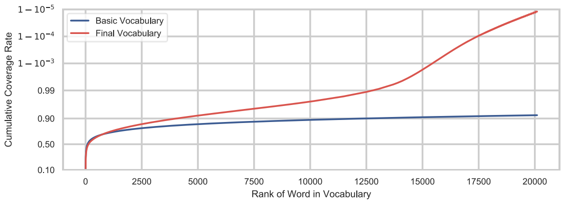


Figure 1: Comparison of the cumulative coverage rate of the vocabulary. The original vocabulary produced by CWS (colored in blue) has a total size of more than 3 million, and the top 20,000 words in the vocabulary has a coverage rate of only 90%. With the proposed framework, the vocabulary coverage rate can be increased to nearly 100% (colored in red) while maintaining a suitable vocabulary size (less than 20,000).

Following the pre-training/fine-tuning paradigm, we collect about 3,500 published books of Chinese novels as the pre-training corpus, aiming at Chinese literary language modeling. Then we collect about 220,000 passages of Chinese modern poetry and lyrics as the fine-tuning corpus. To build the vocabulary, most researchers directly apply frequency-based tokenization methods such as byte pair encoding (BPE, Gage, 1994; Sennrich et al., 2016) or unigram language model (Unigram LM, Kudo, 2018) without considering the Chinese word segmentation (CWS) issue, which is a critical Chinese NLP task and can be modeled by supervised learning (Liu et al., 2014; Yan et al., 2020; Qiu et al., 2020; Duan and Zhao, 2020). Despite these advances, it is still difficult to directly deploy CWS into the tokenization process due to the "long-tail" issue. Figure 1 illustrates the cumulative coverage rate of the vocabulary (sorted by the word frequency) of our corpus produced by the CWS software `THULAC` (Sun et al., 2016). It generates a large vocabulary (containing about 3.6 million words) with a very long low-frequency "tail", which grows slowly to the coverage rate of the vocabulary and takes a nonnegligible portion of coverage. In our case, the top 20,000 words take about 90% coverage on the corpus, and the remaining 3.6 million minus 20,000 words take the remaining 10% coverage, which satisfies Zipf's law (Zipf, 1949). Truncating the "tail" would result in a low coverage rate and too many "unknown" tokens.

2.2 Proposed CWS Framework

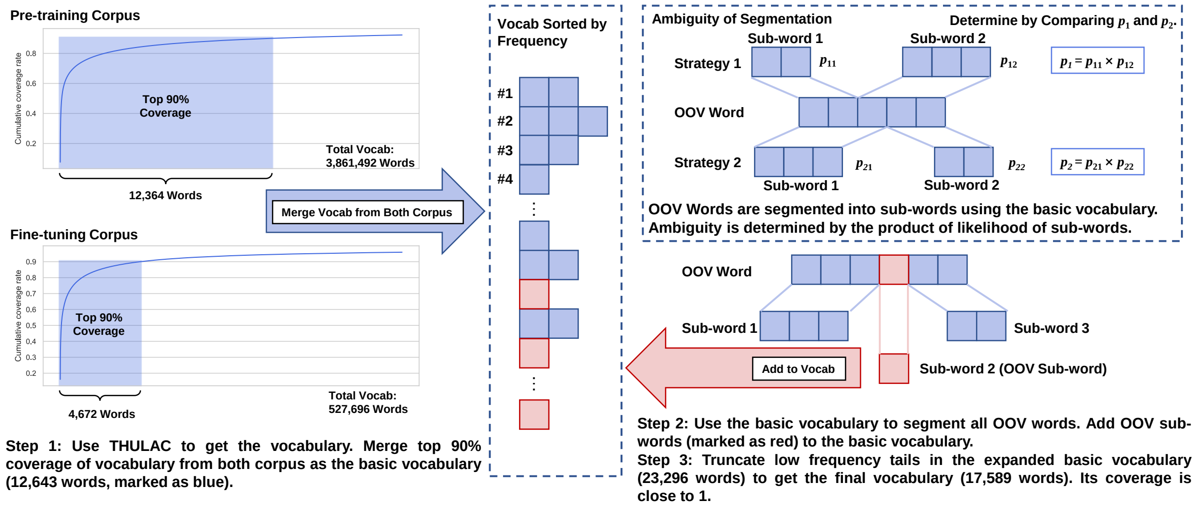
To handle the "long-tail" issue, we propose the following heuristic algorithm, which segments the "tail" into "subwords" using words from the top portion of the vocabulary.

Step 1: Use the CWS tool `THULAC` to process the corpus and acquire the vocabulary, sort it by the word frequency, and choose a top portion of the vocabulary as the *basic vocabulary*.

Step 2: For the out-of-vocabulary (OOV) words, use the *basic vocabulary* to segment them into *subwords*. For subwords outside the basic vocabulary, add them to the basic vocabulary. If an OOV word has different segmentation strategies, determine by choosing the largest likelihood product of subwords.

Step 3: Sort the *expanded* basic vocabulary, and choose a top portion as the *final vocabulary*.

Let V denote the vocabulary of the corpus produced by the CWS model. For each word $w \in V$, its frequency and likelihood are denoted by $n(w)$ and $p(w)$, which satisfy $p(w) = n(w)/\sum_{w \in V} n(w)$. Sort V by $p(w)$, then choose the top portion of the vocabulary with cumulative coverage rate being P_1 (referred to as "top P_1") to construct the basic vocabulary V_{BASIC}. The rest of words in V ("tail") are denoted by $\overline{V}_{BASIC}$. To process $\overline{V}_{BASIC}$, segment each word in $\overline{V}_{BASIC}$ using words in V_{BASIC}. During this process, a word might have different segmentation strategies. To solve the segmentation ambiguity, let $S(w) = \{w_1, w_2, ...\}, w_i \in V_{BASIC}$ denote a segmentation of word $w \in \overline{V}_{BASIC}$ with an ordered sequence of token w_i. We choose the segmentation strategy with the largest likelihood product,



OOV Words are segmented into sub-words using the basic vocabulary. Ambiguity is determined by the product of likelihood of sub-words.

Step 1: Use THULAC to get the vocabulary. Merge top 90% coverage of vocabulary from both corpus as the basic vocabulary (12,643 words, marked as blue).

Step 2: Use the basic vocabulary to segment all OOV words. Add OOV sub-words (marked as red) to the basic vocabulary.

Step 3: Truncate low frequency tails in the expanded basic vocabulary (23,296 words) to get the final vocabulary (17,589 words). Its coverage is close to 1.

Figure 2: Illustration of the proposed CWS framework for Chinese modern poetry.

i.e., $\text{argmax}_{S(w)} \prod_{w \in S(w)} p(w)$. If any word outside V_{BASIC} is found in the segmentation, add it to V_{BASIC}. After all words in $\overline{V}_{BASIC}$ are processed, V_{BASIC} will be expanded to cover all the corpus. Sort the expanded V_{BASIC} by its updated word likelihood and choose the top P_2 of the vocabulary as the final vocabulary V_{FINAL}. The process is described in Algorithm 1.

Algorithm 1 The proposed CWS Framework

```
Input: Training Corpus
Output: $V_{FINAL}$
Acquire $V$ by THULAC, sort $V$ by $p(w)$, choose top $P_1$ of $V$ as $V_{BASIC}$, and its complement as $\overline{V}_{BASIC}$.
for each $w \in \overline{V}_{BASIC}$ do
    Find all possible segmentations $\{S(w)\}$
    $S^*(w) \leftarrow \text{argmax}_{S(w)} \prod_{w \in S(w)} p(w)$
    for each $w^* \in S^*(w)$ do
        Add $w^*$ to $V_{BASIC}$ if $w^* \notin V_{BASIC}$
Update $p(w)$. Sort $V_{BASIC}$ by $p(w)$, choose top $P_2$ of $V_{BASIC}$ as $V_{FINAL}$
return $V_{FINAL}$
```

After all words in $\overline{V}_{BASIC}$ are processed, all sub-words outside V_{BASIC} will be added to the vocabulary. In this way, V_{BASIC} will be expanded to cover all information in the corpus, and all OOV words can be segmented into in-vocabulary sub-words. All information will be kept during this process with no "unknown" tokens. And in the last step, only sub-words with a very low word frequency that are rarely used will be filtered out and replaced with "unknown" tokens. In this way, the final vocabulary will achieve a high coverage rate with a suitable size. The entire process is shown in Figure 2. It combines the CWS model with frequency-based tokenization methods and can generate a vocabulary with suitable size and high coverage rate. In our case, the coverage rate of the final vocabulary is close to 1.0 (larger than $1 - 10^{-4}$, see Figure 1), with a vocabulary size being 17,589.

Since we have two different training corpora, we seek to leverage the word frequency feature from both corpora. For **Step 1**, we choose the top 90% of the vocabulary from a) both corpora and b) the fine-tuning corpus only, then merge them as the basic vocabulary. This emphasizes the fine-tuning corpus to benefit its generation task. The size of basic vocabulary merged in **Step 1** is 12,643. After **Step 2**, the vocabulary size is expanded to 23,296. In **Step 3**, we drop words with a frequency lower than a) 100 on both corpora and b) 10 on the fine-tuning corpus and acquire the final vocabulary with 17,589 words. By observation, the dropped words in **Step 3** are all extremely rare single-length Chinese words.

3 The Diversity-aware Sampling

We train an auto-regressive Transformer language model on the two collected corpus for generation (see the appendices for the model details). As is widely acknowledged, the decoding module plays a crucial role in neural text generation. Stochastic sampling methods such as nucleus sampling (Holtzman et al., 2020), top-*k* sampling (Fan et al., 2018; Holtzman et al., 2018) and temperature sampling can generate texts with higher diversity than traditional decoding methods such as beam search. Recent advances such as MIROSTAT (Basu et al., 2021) and typical sampling (Meister et al., 2023) use adaptive filtering on top of these methods. However, in our model, we observe that traditional algorithms occasionally generate boring and repetitive poetry with low diversity, thus severely hurting the quality of generated poetry.

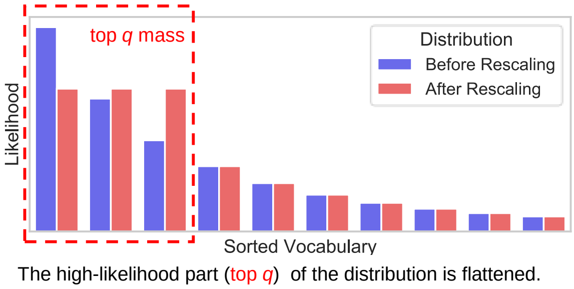


Figure 3: Illustration of the proposed NS-FH algorithm, which flattens and rescales the high-likelihood part of the predicted distribution to emphasize less likely words and increase the diversity of generated poetry.

Recent research has revealed that the low-diversity issue can stem from the high-likelihood part of the predicted distribution (Holtzman et al., 2020; Basu et al., 2021), which is not fully leveraged by traditional sampling methods. We consider the intuition for poetry composing that fluent poetry with too many high-likelihood words can be boring (i.e., the boredom trap, Basu et al., 2021); in contrast, semantically ambiguous poetry with surprising and low-likelihood words can be creative and poetic (e.g., poems by James Joyce or Marcel Proust). Inspired by this, we propose a novel sampling algorithm to emphasize the less likely words on the predicted distribution to increase the diversity. We leverage the notion of nucleus sampling (NS) by defining an additional filtering parameter denoted by q to identify the high-likelihood part ("head") of the vocabulary, denoted by V^{head}. Then we propose to *flatten and evenly redistribute* the probability mass for V^{head}, to emphasize the "comparatively low likelihood" words in the "head". For the low-likelihood part of the distribution ("tail"), we adopt nucleus sampling with parameter p ($p \geq q$) to truncate the "tail" like the tradition. Stochastic sampling is conducted on the flattened and rescaled distribution for all sampling steps. The above method is referred to as *nucleus sampling with flattened head* (NS-FH) algorithm. The diversity gain of the algorithm is controlled by q, which determines the boundary of V^{head}. The algorithm is illustrated in Figure 3.

The proposed algorithm has a close relationship with nucleus sampling. Their truncation mechanisms on the predicted distribution are identically based on the threshold $\{p, q\}$ of the cumulative probability density function. And our method features in using an additional threshold q to determine another smaller portion of the predicted distribution and flattening their probability density function. And since q is also a threshold for the cumulative probability density function, the flattening manipulation will not affect low-entropy distributions that only contain an "unquestionably correct" word which takes most of the probability mass, thus not hurting the fluency of the generated poetry. We later show in the empirical results that the proposed method can closely follow the behavior of nucleus sampling when using a small value of q while exhibiting higher diversity when setting a large value of q, achieving diversity-aware sampling.

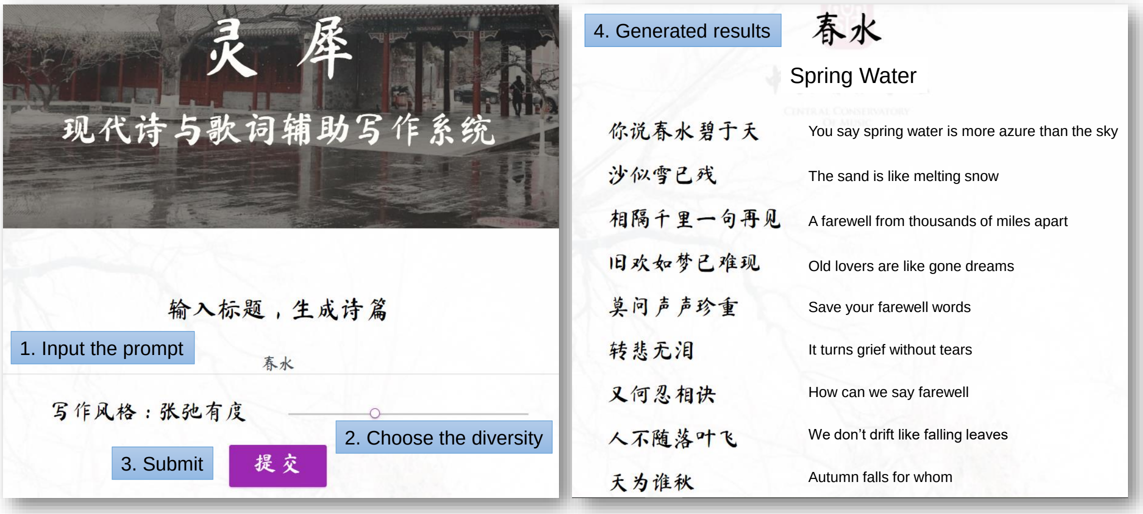
4 Demonstration and Evaluation

4.1 System Interface

The system interface and poetry example are shown in Figure 4. We provide a plain mode and an advanced mode. For the plain mode, input the poetry prompt, choose the diversity parameter with the slider, and hit the "submit" button to get a generated poetry. By default, the system uses the proposed NS-FH algorithm as the decoding method. In the advanced mode, we implemented the three golden sampling methods (nucleus sampling, top-k sampling, and temperature sampling) as well as two novel sampling methods (MIROSTAT by Basu et al., 2021 and typical sampling by Meister et al., 2023) for users to choose and compare.

4.2 Impact of Sampling Algorithms

It is noteworthy that the performance of the sampling algorithm is highly dependent on the sampling hyperparameter. To illustrate and compare their impact on the generated poetry, we take advantage of the quality-diversity trade-off feature (Nadeem et al., 2020; Zhang et al., 2021; Basu et al., 2021). By tuning the hyperparameter for one sampling algorithm, the quality (fluency) metric and the diversity metric of the generated poetry form a trade-off curve in which an increase in the diversity metric will decrease the fluency metric.



春水

Spring Water

你说春水碧于天	You say spring water is more azure than the sky
沙似雪已残	The sand is like melting snow
相隔千里一句再见	A farewell from thousands of miles apart
旧欢如梦已难现	Old lovers are like gone dreams
莫问声声珍重	Save your farewell words
转悲无泪	It turns grief without tears
又何忍相诀	How can we say farewell
人不随落叶飞	We don’t drift like falling leaves
天为谁秋	Autumn falls for whom

Figure 4: The system interface and generated poetry sample of `Lingxi`.

Such a feature demands a holistic picture of the trade-off curve by traversing the hyperparameter space and evaluating automatic metrics on the generated poetry to reveal the nature of the algorithm. As a result, we leverage the trade-off feature by aligning corresponding metrics on a 2D plane throughout our evaluation.

Concretely, we use every poetry title from the validation set as the input prompt for generation, which results in 15,218 generations per sampling algorithm per hyperparameter. We choose the fluency metric as the perplexity (PPL) of generated poetry (lower score indicates higher fluency but more boredom). We then choose the following three diversity metrics: the Zipf coefficient (Zipf, 1949; Newman, 2005) which reflects the sloping tendency of the word frequency distribution (lower score indicates a flatter distribution and higher diversity); the entropy of n-gram distribution (Zhang et al., 2018) which reflects the diversity of n-grams (higher score indicates less repeating n-*grams* and higher diversity); the self-BLEU score (Zhu et al., 2018; Holtzman et al., 2020) which reflects the overlapping tendency among different generated poetry passages (lower score indicates less overlapping and higher diversity). We calculate the human-level metrics from the validation set as the reference point.

Results for the fluency-diversity trade-off curves are shown from Figure 5 to Figure 7. We show regions around the human (reference) metric point for a clear view (see Figure 10 to Figure 14 in the appendix for full curves). Results show that the human-level metric of Chinese modern poetry suggests a **high diversity requirement** (low Zipf coefficient, high entropy of n-gram, and low self-BLEU score). To achieve human-level diversity, traditional methods must greatly relax their sampling parameter and almost degrade to pure sampling (sampling on the original predicted distribution with no filtering) for maximum diversity. By contrast, our method controls the trade-off curve by tuning the diversity parameter q without fully relaxing the filtering. When using a small value of $q = 0.20$, our method closely follows the trajectory

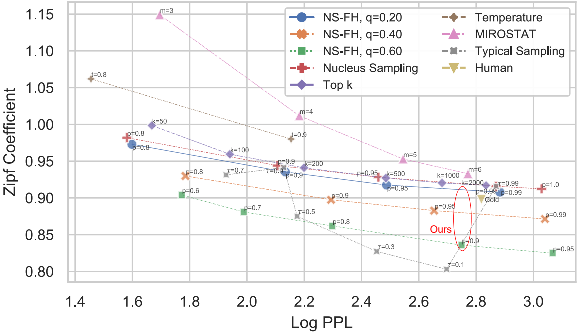


Figure 5: The trade-off curve for Zipf coefficient (diversity ↓) against perplexity. Although all methods loosely converge to the human (reference) point, traditional methods must greatly relax their sampling parameters ($p = 0.99$, $k = 2000$, $t > 0.9$, $m > 6$, $\tau = 0.99$), i.e., almost degrade to pure sampling to achieve human-level metrics. With a small diversity parameter, our method (NS-FH, q=0.20) has a similar trajectory to nucleus sampling; with a large diversity parameter, our method (NS-FH, q=0.60) exhibits higher diversity (lower curve) than other methods in most cases (only except for typical sampling with $\tau = 0.10$), achieving diversity-aware sampling.

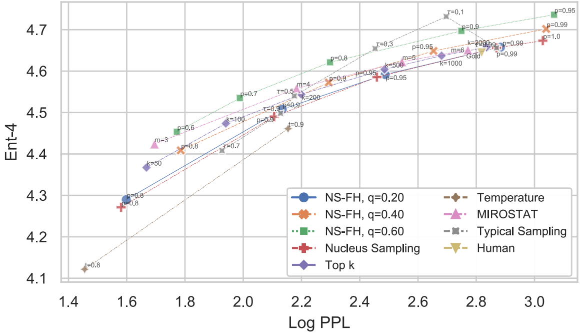


Figure 6: The trade-off curve for the entropy of 4-gram distribution (Ent-4, diversity ↑) against perplexity. All methods exhibit close trajectories with each other and converge to the human (reference) point. Our method (NS-FH, q=0.60) can also achieve the diversity upper bound achieved by typical sampling with $\tau = 0.1$, meanwhile its trade-off curve still stays close to the reference point.

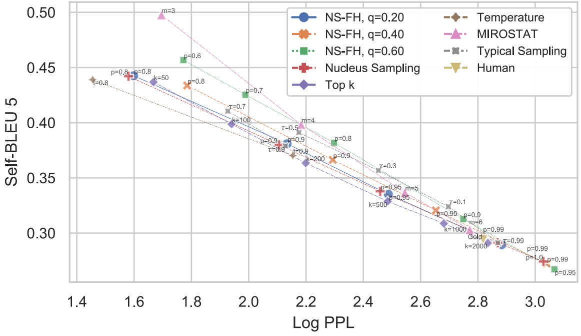


Figure 7: The trade-off curve for self-BLEU 5 (diversity ↓) against perplexity. While all methods exhibit close trajectories near the human (reference) point, our method (NS-FH, q=0.60) achieves the highest diversity boundary (lowest metric in the lower right region).

of nucleus sampling and converges to the reference point; when using a large value of $q = 0.60$, our method achieves a comparable or higher diversity metric to traditional methods without deviating from the reference point. These results indicate that our method can achieve diversity-aware sampling with good metric performance.

Note a very interesting behavior of the NS-FH algorithm that by setting $q = 0.60$ (colored in green), i.e., the top 60% of cumulative probability mass of the predicted distribution is completely flattened and ignored, the trade-off curve does not drift away from the human-level metric point (see the *performance degradation* of sampling algorithms illustrated in Figure 3 by Nadeem et al., 2020), while even achieving higher diversity metrics and boundaries in most cases. This suggests that in our task with a high requirement for diversity, the probability of high-likelihood words on a “flat” distribution with high entropy can be *ignored*. Completely flattening the distribution and ignoring the predicted probability of high-likelihood words can counter-intuitively achieve comparable or better results with higher diversity. By inference, such a method might be suitable for other artistic generation tasks like music or drawings that require high diversity.

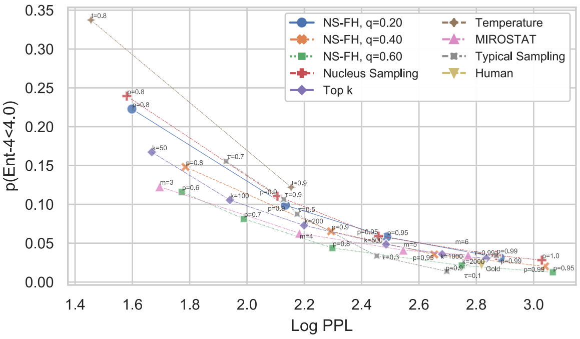


Figure 8: The trade-off curve for $p(\text{Ent-4} < 4.0)$ (↓) against perplexity. Similar to the results of previous diversity metrics, our method achieves a lower metric boundary than traditional methods (except for typical sampling with $\tau \leq 0.3$) and still converges to the human (reference) point.

Also, note the stochastic nature of the sampling methods, which yields a stochastic trajectory of tokens with variational quality. Under the same sampling parameter and conditions, the generated poetry might occasionally be repetitive and wordy. So we focus on the distribution of the entropy of n-gram metric to investigate the repetition tendency. We calculate $p(\text{Ent-4} < \eta)$, which reflects the chance that the generated poetry has the entropy of n-grams lower than the threshold η. Empirically, we set $\eta = 4$ to get a meaningful observation. Results are shown in Figure 8. They show that naive sampling parameters for traditional methods easily result in more repetition, e.g., nucleus sampling with $p = 0.80$ has $p(\text{Ent-4} < 4) \approx 0.24$. Similar to previous results, traditional methods have to greatly relax the filtering parameter or degrade to pure sampling to approach human-level repetition tendency. By contrast, our method with $q = 0.60$ achieves a lower curve (less repetition chance) than most methods and also stay close to the reference point. This indicates that our method with a high diversity parameter can grant less repetition as well as maintain a close relationship of repetition tendency to human behavior without fully relaxing the filtering.

4.3 Rhyming Feature

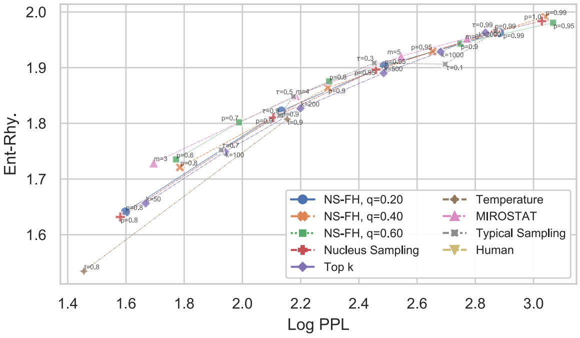


Figure 9: The trade-off curve for the entropy of rhyming word distribution against the fluency metric. Results show that our method has on-par rhyming performance with all traditional methods.

We also consider an additional metric to reflect the rhyming feature of the generated poetry. Similar to the entropy metric, we calculate the entropy of rhyming word distribution by $\mathbb{E}\{-\log p_{rhyme}(x)\}$, where $p_{rhyme}(x)$ is the rhyming word frequency distribution (higher score indicates higher diversity but less rhymed). Similarly, we plot the fluency-rhyming trade-off curves in Figure 9. The results show that our method achieves on-par rhyming performance with all baseline methods. It indicates that the flattening manipulation of our method does not hurt the rhyming feature of the generated poetry, despite that our method flattens the top portion of the cumulative probability mass of the predicted distribution.

Since rhyming is an important feature for composing Chinese modern poetry, we provide an additional function to control the rhyme of the generated poetry. In the advanced mode of `LingXi`, we adopt a re-ranking and replacing mechanism for rhyming. During the generation process, once a [`NEWLINE`] symbol was generated, we try to replace its previous token with a rhyming one. We create hypotheses by combining each token from the vocabulary that fits the requirement of the chosen rhyme with a [`NEWLINE`] token, recalculate the average perplexity of the combined tokens (hypothesis), and choose the hypothesis with the lowest average perplexity to replace the generated one. One issue is that there might be cases in which all hypotheses have high perplexity (low likelihood). So we set an additional threshold that the hypothesis must have an average perplexity lower than the threshold to trigger the replacement. If none of the hypotheses meet the requirement of the threshold, the generation remains unrevised. The above method can achieve a trade-off between rhyming and fluency by tuning the perplexity threshold. Empirically, we set the threshold to 50 to achieve a good performance. Although the method cannot guarantee that all generated lines of poetry meet the rhyming requirement, it features in its flexibility to plug into all decoding methods implemented in our system without modifying the language model or the algorithms.

5 Conclusion

We present `Lingxi`, a diversity-aware Chinese modern poetry generation system. To address the CWS issue, we propose a novel framework that combines CWS with the frequency-based method, which can create a vocabulary with a suitable size and high coverage. To increase the diversity of generated poetry, we propose nucleus sampling with flattened head (NS-FH) algorithm which achieves controllable diversity with on-par or better performance compared to traditional sampling methods. The proposed sampling algorithm provides a new approach to increase the diversity of neural text generation via the decoding module, which might be beneficial for artistic generation cases that have high requirements for the diversity or novelty.

Acknowledgments

This work has been funded by: (i) Special Program of National Natural Science Foundation of China (Grant No. T2341003), (ii) Advanced Discipline Construction Project of Beijing Universities, (iii) Major Program of National Social Science Fund of China (Grant No. 21ZD19), (iv) Nation Culture and Tourism Technological Innovation Engineering Project (Research and Application of 3D Music), (v) National Philosophy and Social Science Project (Grant No. 22VJXG012).

References

Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. 2021. MIROSTAT: A neural text decoding algorithm that directly controls perplexity. In *International Conference on Learning Representations*.

Sufeng Duan and Hai Zhao. 2020. Attention is all you need for Chinese word segmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3862–3872, Online. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Philip Gage. 1994. A new algorithm for data compression. *C Users J.*, 12(2):23–38.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1638–1649, Melbourne, Australia. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Hsin-Pei Lee, Jhih-Sheng Fang, and Wei-Yun Ma. 2019. iComposer: An automatic songwriting system for Chinese popular music. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 84–88, Minneapolis, Minnesota. Association for Computational Linguistics.

Yijia Liu, Yue Zhang, Wanxiang Che, Ting Liu, and Fan Wu. 2014. Domain adaptation for CRF-based Chinese word segmentation using free annotations. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 864–874, Doha, Qatar. Association for Computational Linguistics.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023. Locally Typical Sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121.

Moin Nadeem, Tianxing He, Kyunghyun Cho, and James Glass. 2020. A systematic characterization of sampling algorithms for open-ended language generation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 334–346, Suzhou, China. Association for Computational Linguistics.

Mark EJ Newman. 2005. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351.

Xipeng Qiu, Hengzhi Pei, Hang Yan, and Xuanjing Huang. 2020. A concise model for multi-criteria Chinese word segmentation with transformer encoder. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2887–2897, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Maosong Sun, Xinxiong Chen, Kaixu Zhang, Zhipeng Guo, and Zhiyuan Liu. 2016. Thulac: An efficient lexical analyzer for chinese. `http://thulac.thunlp.org/`.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. A graph-based model for joint Chinese word segmentation and dependency parsing. *Transactions of the Association for Computational Linguistics*, 8:78–92.

Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021. Trading off diversity and quality in natural language generation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 25–33, Online. Association for Computational Linguistics.

Le Zhang, Rongsheng Zhang, Xiaoxi Mao, and Yongzhu Chang. 2022. QiuNiu: A Chinese lyrics generation system with passage-level input. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 76–82, Dublin, Ireland. Association for Computational Linguistics.

Rongsheng Zhang, Xiaoxi Mao, Le Li, Lin Jiang, Lin Chen, Zhiwei Hu, Yadong Xi, Changjie Fan, and Minlie Huang. 2020. 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.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '18, pages 1097–1100, New York, NY, USA. Association for Computing Machinery.

George K. Zipf. 1949. *Human Behaviour and the Principle of Least Effort*. Addison-Wesley.

A Model Configuration, Training Details and Supplementary Results

Detailed configurations and parameters of our model are listed in Table 1. For special tokens, the [TITLE] token is added directly after the title of each poetry passage in the fine-tuning corpus to capture the title feature. The [START-OF-PASSAGE] token is added before the starting token of the first poetry line. We create a replica for each poetry passage excluding the title and [TITLE] token, and mix them with the original corpus as data augmentation. The [NEWLINE] token is added at the end of each poetry line in replace of the newline character, and the [END-OF-PASSAGE] token is added at the end of each poetry passage. English words and letters are assigned [UNK-EW] tokens. Other unknown words and sub-words are assigned [UNK] tokens. For poetry passages longer than the maximum context length, we create training samples using a sliding window with stride being half of the maximum context length. We split train/validation/test sets using the common ratio of 85%/7%/8% (token ratio for the pre-training corpus, passage ratio for the fine-tuning corpus).

The model is an auto-regressive Transformer decoder, using cross entropy as the training loss. The training process is developed using the Huggingface library by Wolf et al. (2019). It achieves monotonic convergence of perplexity (PPL) on the validation set of the pre-training corpus at the end of the pre-training steps. We choose the best fine-tuning epoch of the model with the lowest PPL on the validation set of the fine-tuning corpus as the final model. For the generated poetry, their full metric trade-off curves for all sampling hyperparameters and sampling algorithms are shown from Figure 10 to Figure 14.

Parameters	Value
number of Transformer layers	24
number of Transformer attention heads	16
embedding size	1,024
vocabulary size	17,589
maximum context length	128
number of network parameters	330 million
pre-training epochs	20
fine-tuning epochs	10
batch size per GPU	32
number of training GPUs	8 NVIDIA® GeForce® RTX 2080 Ti
pre-training learning rate	2×10^{-4}
fine-tuning learning rate	$\{1 \times 10^{-5}, 2 \times 10^{-5}\}$
learning rate decay	linear decay
warm-up steps	1% of total steps
optimizer	Adam optimizer (Kingma and Ba, 2014)
weight decay	0.01
PPL on validation set after pre-training	17.58
best fine-tuning epoch	epoch 4, learning rate being 2×10^{-5}
best PPL on fine-tuning validation set	16.75
full sampling hyperparameter space for the trade-off curves from Figure 10 to Figure 14.	$p \in \{0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99, 1\}$ $q \in \{0.2, 0.4, 0.6\}, p \geq q$ $k \in \{10, 20, 50, 100, 200, 500, 1000, 2000\}$ $t \in \{0.6, 0.7, 0.8, 0.9, 1.1\}$ $m \in \{2, 3, 4, 5, 6\}$ $\tau \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99\}$

Table 1: Model configuration, training details and sampling parameters

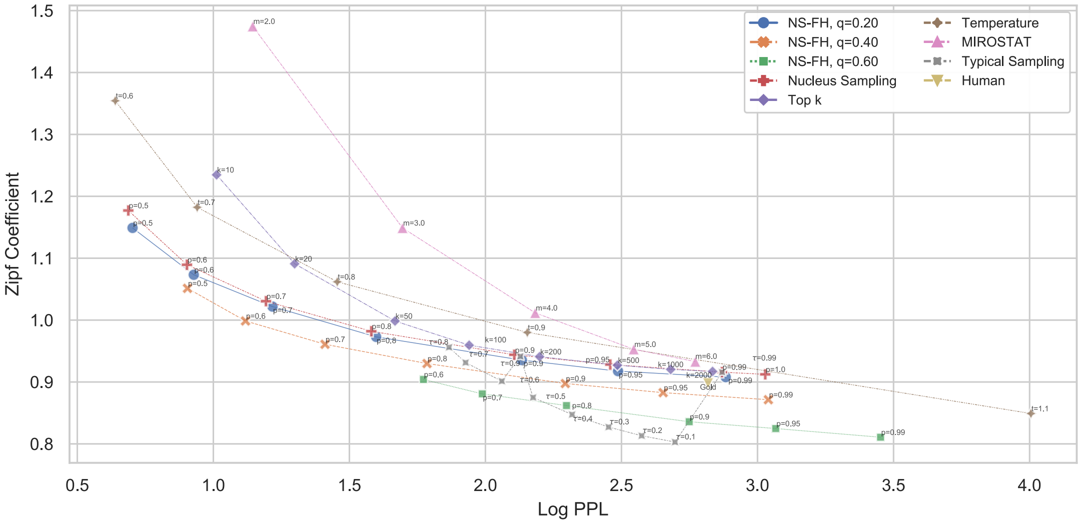


Figure 10: Full trade-off curve for Zipf coefficient against perplexity.

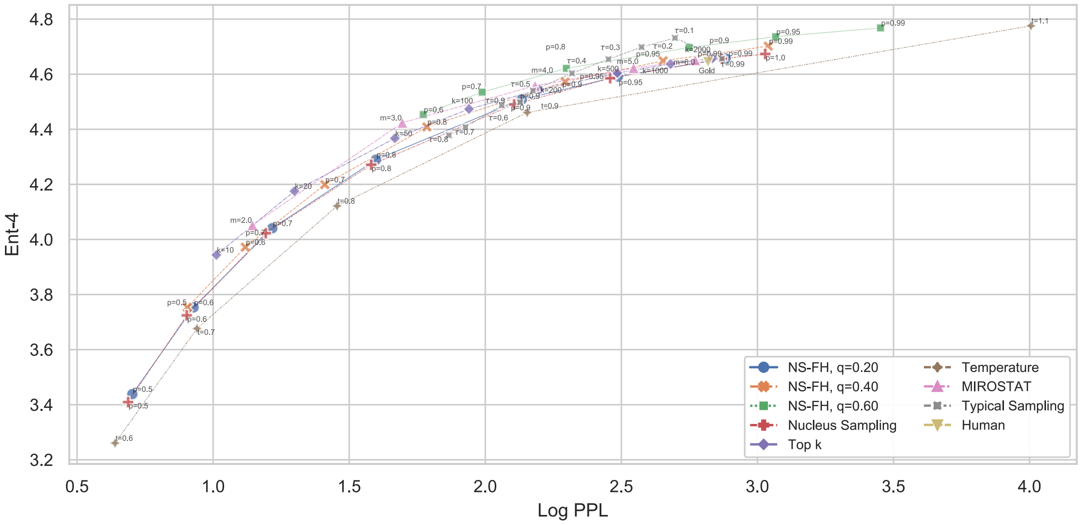


Figure 11: Full trade-off curve for the entropy of 4-gram distribution (Ent-4) against perplexity.

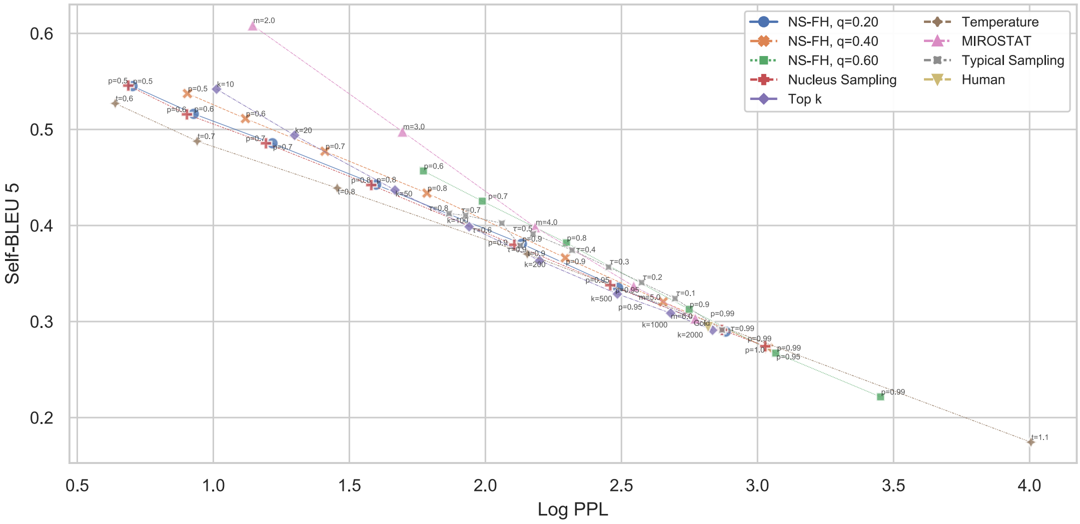


Figure 12: Full trade-off curve for self-BLEU 5 against perplexity.

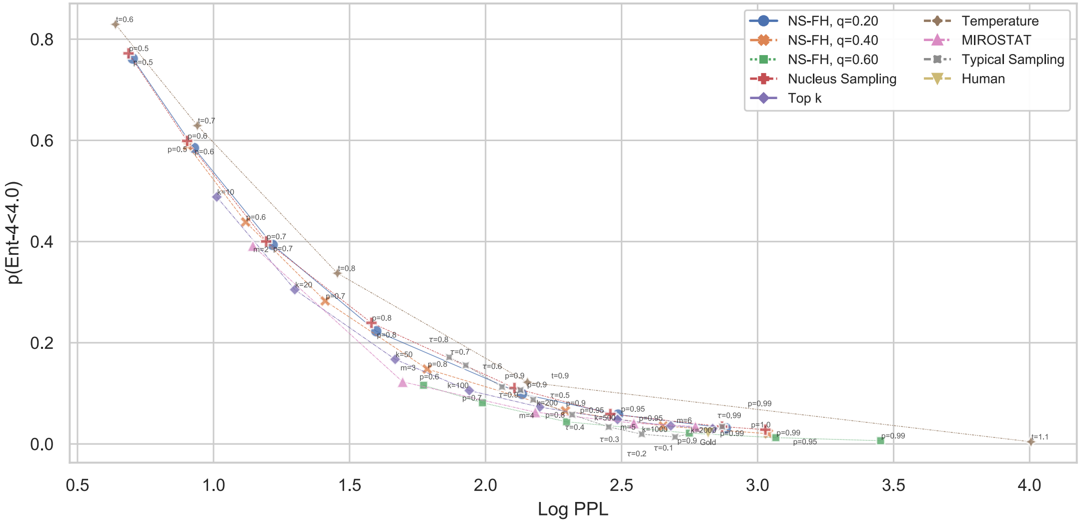


Figure 13: Full trade-off curve for $p(\text{Ent-4} < 4.0)$ against perplexity.

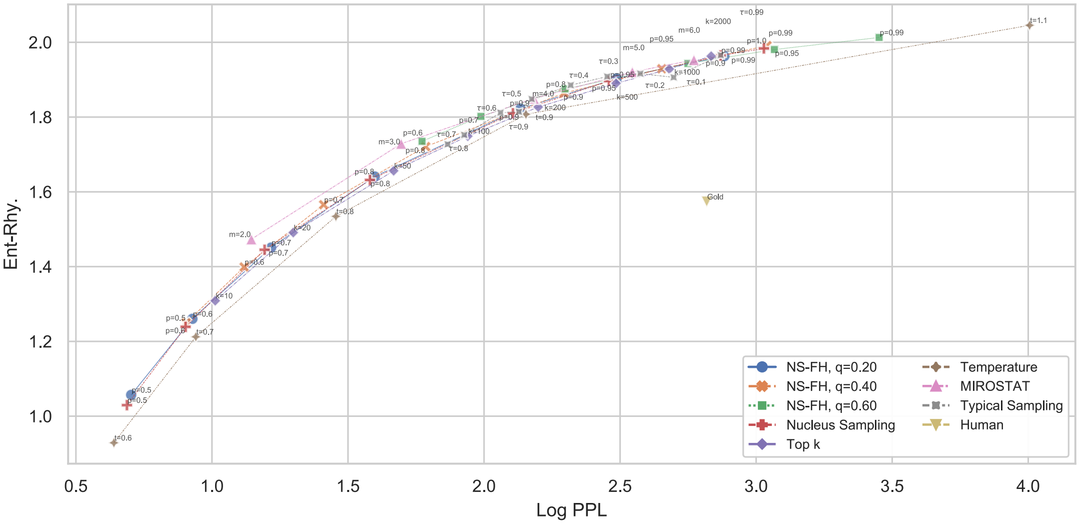


Figure 14: Full trade-off curve for the entropy of rhyme distribution against the fluency metric.