LYRA: Automatic Lyrics Annotation for Russian Rap using Retrieval and Generation

Kirill Anpilov

17 января 2026 г.

Аннотация

Automatic annotation and explanation of song lyrics presents a significant challenge in NLP, especially for Russian rap which is rich with metaphors, cultural references, and wordplay. I present LYRA (Lyrics Retrieval and Annotation), a system for generating explanations of Russian rap lyrics. The dataset consists of 22,220 annotations collected from Genius for 3,291 Russian rap songs. I implement and compare multiple retrieval approaches: TF-IDF (ROUGE-1: 0.0070), BM25 (ROUGE-1: 0.0140), SBERT semantic search (ROUGE-1: 0.0087), Hybrid retrieval (ROUGE-1: 0.0102), and Ensemble methods (ROUGE-1: 0.0104). I also compare multilingual embedding backbones and observe improvements with E5-multilingual (ROUGE-1: 0.0102). The best retrieval method is BM25, which outperforms TF-IDF by 2x on ROUGE-1. I additionally evaluate a generative approach based on ruT5 and a retrieval-augmented variant (BM25 + ruT5), which substantially improve overlap metrics compared to retrieval-only baselines.

Keywords: lyrics annotation, explanation generation, Russian NLP, retrieval methods, transformer models

1 Introduction

Music and poetry are vehicles for complex human expression, densely packed with metaphors, cultural allusions, and emotional nuance. In Russian rap and hip-hop, lyrical complexity is particularly pronounced, with artists weaving sophisticated wordplay, historical references, and literary allusions into their verses.

Understanding these lyrics often requires knowledge beyond surface-level language comprehension:

- **Metaphorical expressions** ("Я вижу город под подошвой" - expressing dominance over one's environment)
- **Cultural and historical references** ("Петербург - это Ленинград в противогазе" - evoking Soviet history)

- **Wordplay and double meanings** ("Играю минор, но это мой мажор" - puns on musical and emotional registers)

Manual annotation of lyrics is labor-intensive and requires both linguistic knowledge and cultural understanding. I propose LYRA, a system that automatically annotates song lyrics using retrieval-based methods (TF-IDF, BM25, SBERT), hybrid methods, ensemble approaches, and transformer-based generation (ruT5). In experiments on a 2,000-example subset, BM25 is the strongest retrieval baseline (ROUGE-1: 0.014), while ruT5 + RAG reaches ROUGE-1: 0.279 and substantially improves overlap metrics.

1.1 Team

Kirill Anpilov developed all components of the LYRA system: dataset collection, implementation of all retrieval and generation approaches, evaluation framework, and this report.

1.2 Contributions

My main contributions are:

1. A dataset of 22,220 annotations for 3,291 Russian rap songs from Genius
2. Implementation of retrieval approaches: TF-IDF, BM25, SBERT, Hybrid, and Ensemble
3. Evaluation protocol on a 2,000-example subset with ROUGE metrics
4. Analysis of approach strengths and limitations for lyrics annotation
5. A reproducible codebase with notebooks and evaluation logs

2 Related Work

2.1 Lyrics Analysis and NLP

The application of NLP to music lyrics has a growing body of work. Recent surveys summarize lyrics processing as a full-stack area that spans analysis, generation, and downstream applications such as recommendation (e.g., Watanabe and Goto, 2020). These works motivate lyric-specific preprocessing (short lines, slang, non-standard grammar) and highlight the importance of lexical and semantic features.

2.2 Automated Lyric Annotation

Sterckx et al. (2017), *Break it Down for Me: A Study in Automated Lyric Annotation*, introduce the task of automated lyric annotation (ALA) and release a large dataset of crowdsourced Genius annotations. They define ALA as rewriting lyric lines into clearer explanations while adding contextual knowledge when needed. The dataset contains 803,720 lyric-annotation pairs filtered from Genius and is described in detail in their paper (`https://arxiv.org/abs/1708.03492`). The work also references public lyric databases such as MetroLyrics and Genius (`https://genius.com`). For baselines, they compare SMT, Seq2Seq, and retrieval approaches, and evaluate with BLEU, METEOR, and SARI, plus human ratings. This is the closest prior art to my setting and motivates the retrieval and generation baselines I implement.

2.3 Metaphor and Figurative Language Detection

Understanding lyrics requires recognizing figurative language. The VU Amsterdam Metaphor Corpus and the VUA Metaphor Detection shared tasks (NAACL 2018, ACL 2020) provide large-scale benchmarks for metaphor detection. This work goes beyond detection to explanation: the goal is not just to identify metaphors, but to explain their meaning in context.

2.4 Explanation Generation

Explanation generation is commonly formulated as a text-to-text task, where models produce natural-language rationales or interpretations conditioned on input text. Datasets such as e-SNLI (explanations for NLI), CoS-E (commonsense explanations), and ComVE (commonsense validation and explanation) established standard evaluation protocols for explanation quality. Sequence-to-sequence architectures (e.g., T5-like models) are a standard choice for this setup.

2.5 Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) combines search over external examples with conditional generation. RAG-style systems and dense retrieval methods (e.g., DPR) show that a retrieved memory can improve factuality and grounding. This is particularly relevant for lyrics, where retrieved annotations can provide cultural or contextual grounding for more faithful explanations.

2.6 Transformer Models for Russian

For Russian, encoder and encoder-decoder transformer models are widely used for semantic similarity and generation tasks. Russian BERT-like encoders are commonly used for retrieval, while ruT5 and multilingual T5 variants enable fluent explanation generation. Multilingual sentence embedding models (e.g., E5-style encoders) provide robust semantic matching across noisy and short texts.

2.7 Distinction from Related Work

While prior work addresses metaphor detection and explanation in other domains, this work specifically targets song lyrics in Russian, which poses unique challenges:

1. **Poetic license**: Non-standard grammar and word order
2. **Cultural density**: Heavy use of historical and literary references
3. **Dataset availability**: Large-scale annotated lyrics corpus

2.8 Competitive Approaches

Based on prior work and common baselines in explanation generation, I consider the following competitor approaches:

- **Lexical retrieval**: TF-IDF or BM25 with cosine similarity, returning the annotation of the closest fragment.
- **Semantic retrieval**: sentence embeddings (SBERT, multilingual encoders) with nearest-neighbor search.
- **Hybrid and ensemble retrieval**: weighted combinations of lexical and semantic scores.
- **Abstractive generation**: encoder-decoder models (ruT5) trained to generate explanations.
- **RAG**: retrieval-augmented generation, where retrieved annotations are injected into the prompt.
- **Upper-bound alternatives**: large LLMs in zero-shot mode or human explanations (not evaluated here).

3 Dataset

3.1 Data Collection

Annotations are collected from Genius (genius.com), a crowd-sourced music annotation platform. Genius annotations are user-created explanations of song lines, including:

- Explanations of metaphors and wordplay
- Historical or cultural context

- References to other songs or artists
- Artist interviews and background

Collection Process:

1. Selected popular Russian rap artists (Pharaoh, Miyagi, Скриптонит, and others)
2. Searched for their songs on Genius using the Genius API
3. Retrieved all annotations (referents) for each song
4. Collected metadata: artist, title, votes, fragment, annotation
5. Final dataset: 3,291 songs with 22,220 annotations

3.2 Dataset Statistics

Metric	**Value**
Number of songs	3,291
Total annotations	22,220
Average annotations per song	6.75
Average fragment length (words)	10.84
Average annotation length (words)	36.12
Vocabulary size (fragments)	65,663
Vocabulary size (annotations)	168,530

Таблица 1: Dataset statistics for LYRA corpus

3.3 Evaluation Setup

For all experiments, I use:

- **Subset size**: 2,000 annotations (randomly sampled with seed=42)
- **Evaluation**: Leave-one-out cross-validation
- **Metrics**: ROUGE-1, ROUGE-2, ROUGE-L, BLEU

This subset ensures computational feasibility while maintaining statistical significance.

4 Model Description

I implement and evaluate multiple retrieval approaches and include a generative extension for lyrics annotation. The solutions include lexical retrieval (TF-IDF, BM25), semantic retrieval (SBERT with multilingual backbones such as E5), hybrid and ensemble retrieval, and generation with ruT5 and a retrieval-augmented variant.

4.1 Baseline: TF-IDF Retrieval

Method: Traditional information retrieval using Term Frequency-Inverse Document Frequency.

Algorithm:

1. Vectorize all fragments using TF-IDF (1,000 features, bigrams)
2. For query fragment, compute cosine similarity to all corpus fragments
3. Return annotation of most similar fragment (excluding self)

Complexity: $O(n \times d)$ where $n =$ dataset size, $d =$ feature dimension.

4.2 BM25 Retrieval

Method: Probabilistic retrieval model, industry standard for search engines.

Key features:

- Improved over TF-IDF with saturation and length normalization
- Parameters: $k_1 = 1.5$ (term frequency saturation), $b = 0.75$ (length normalization)
- Custom tokenization for Russian: regex pattern for Cyrillic and Latin characters

BM25 scoring function:

$$\text{score}(q, d) = \sum_{t \in q} \text{IDF}(t) \cdot \frac{f(t, d) \cdot (k_1 + 1)}{f(t, d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{\text{avgdl}})}$$

where $f(t, d)$ is term frequency in document d, and avgdl is average document length.

4.3 SBERT Semantic Search

Method: Semantic similarity using sentence embeddings.

Model: `paraphrase-multilingual-MiniLM-L12-v2`

Algorithm:

1. Encode all fragments into 384-dimensional embeddings
2. For query, compute cosine similarity in embedding space
3. Return annotation of most similar fragment

Advantage: Captures semantic similarity beyond lexical overlap.

4.4 Hybrid Retrieval

Method: Combines BM25 (lexical) and SBERT (semantic) signals.

Formula:

$$\text{score}_{\text{hybrid}} = \alpha \cdot \text{score}_{\text{BM25}} + (1 - \alpha) \cdot \text{score}_{\text{SBERT}}$$

I test $\alpha \in \{0.3, 0.5, 0.7\}$ and report best result ($\alpha = 0.5$).

4.5 Ensemble Retrieval

Method: Weighted combination of TF-IDF, BM25, and SBERT.

Formula:

$$\text{score}_{\text{ensemble}} = w_1 \cdot \text{score}_{\text{TF-IDF}} + w_2 \cdot \text{score}_{\text{BM25}} + w_3 \cdot \text{score}_{\text{SBERT}}$$

All scores are min-max normalized to $[0, 1]$ before combination.

I test multiple weight configurations and report best: equal weights ($w_1 = w_2 = w_3 = 0.33$).

4.6 ruT5 Generation

Method: Sequence-to-sequence generation using a Russian T5 model.

Model: `ai-forever/ruT5-base` (222M parameters)

Training:

- Input: Song fragment
- Output: Annotation explanation
- Train/val split: 90/10
- Epochs: 3
- Batch size: 8

- Learning rate: 5e-5
- Max source length: 128 tokens
- Max target length: 256 tokens

4.7 ruT5 with RAG

Method: Retrieval-Augmented Generation combining BM25 retrieval and ruT5 generation.

Algorithm:

1. Use BM25 to retrieve top-3 similar fragments and their annotations
2. Construct input: `CONTEXT: [retrieved examples] FRAGMENT: [query] ANNOTATION:`
3. Generate annotation using fine-tuned ruT5

5 Experiments

5.1 Metrics

I use standard text generation metrics:

- **ROUGE-1**: Unigram overlap (recall, precision, F1)
- **ROUGE-2**: Bigram overlap
- **ROUGE-L**: Longest common subsequence

All metrics are computed using `rouge-score`. BLEU is not reported for retrieval baselines due to its low interpretability on this task.

5.2 Experiment Setup

Hardware:

- Retrieval methods: CPU (MacBook Pro M1)
- Generation methods: GPU (NVIDIA Tesla T4, 16GB)

Evaluation protocol:

1. Sample 2,000 annotations randomly (seed=42)
2. For each annotation:
    - Remove it from corpus (leave-one-out)
    - Generate/retrieve annotation for its fragment
    - Compute metrics against ground truth
3. Average metrics across all examples

5.3 Baselines

Random baseline: Randomly select annotation from corpus. Expected ROUGE-1: ~0.002.

TF-IDF: Simple lexical baseline to establish lower bound.

6 Results

6.1 Quantitative Results

Method	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
TF-IDF Retrieval	0.0070	0.0027	0.0062
SBERT Retrieval	0.0087	0.0050	0.0087
Hybrid ($\alpha = 0.5$)	0.0102	0.0050	0.0099
Ensemble (Equal)	0.0104	0.0043	0.0102
BM25 Retrieval	**0.0140**	0.0019	**0.0124**

Таблица 2: Quantitative evaluation results on a 2,000-example subset. Best retrieval method: BM25.

6.2 Generative Model Results

I evaluate ruT5 and a retrieval-augmented variant (BM25 + ruT5) on the same 2,000-example evaluation subset. The generative models substantially outperform retrieval-only baselines in token overlap metrics.

Method	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
ruT5 Generation	0.232	0.206	0.224
ruT5 + RAG (BM25)	0.279	0.251	0.271

Таблица 3: Generative model results on the 2,000-example subset.

Key findings:

1. BM25 is the best retrieval method (+100% over TF-IDF baseline)
2. Ensemble methods provide marginal improvement over single methods
3. Semantic models (SBERT) benefit from stronger multilingual backbones (E5-multilingual)
4. Generative models (ruT5 and ruT5+RAG) outperform retrieval-only approaches by a large margin on ROUGE

6.3 Embedding Backbone Comparison

I evaluate different multilingual sentence encoders for SBERT retrieval. The best result is obtained with `intfloat/multilingual-e5-small` (ROUGE-1: 0.0102), slightly outperforming the default `paraphrase-multilingual-MiniLM-L12-v2`.

6.4 Qualitative Analysis

Example 1 - Metaphor explanation:

Fragment: “Я вижу город под подошвой”

BM25 retrieved: “Метафора превосходства над городом...” (partial match)

Ground truth: “Метафора превосходства над городом, взгляд сверху. Лирический герой чувствует себя выше обыденности мегаполиса.”

Analysis: Retrieval finds a thematically aligned explanation but often lacks paraphrasing and compositionality.

6.5 Discussion

Why retrieval methods plateau:

- Bound by existing annotations in corpus
- Cannot paraphrase or generalize
- Suffer from vocabulary mismatch

Why generation methods may excel:

- Can synthesize novel explanations
- Better at paraphrasing and generalization
- Leverage pre-trained knowledge from ruT5

Why RAG is promising:

- Retrieval provides relevant examples as context
- Generation creates fluent, tailored explanations
- Combines strengths of both paradigms

7 Conclusion

I presented LYRA, a system for automatic lyrics annotation in Russian rap. I implemented and evaluated multiple retrieval approaches and included a generative extension (ruT5 with retrieval augmentation). The results demonstrate:

7.1 Key Findings

1. BM25 is the best retrieval-only method (ROUGE-1: 0.0140)
2. Ensemble methods provide small gains over single models
3. Stronger multilingual encoders (E5) improve semantic retrieval
4. Dataset of 22,220 annotations enables meaningful evaluation
5. Generative models (ruT5, ruT5+RAG) significantly improve ROUGE scores (0.23–0.28 range)

7.2 Contributions

- Russian lyrics annotation dataset (22,220 examples)
- Comparison of retrieval approaches and ensembles
- Evaluation of generative extensions (ruT5, RAG)
- Open-source implementation for reproducibility

7.3 Future Work

1. **Larger models**: Test larger encoder-decoder models or instruction-tuned LLMs
2. **Multi-task learning**: Joint training on metaphor detection and explanation
3. **Cross-lingual**: Extend to English lyrics and compare
4. **Human evaluation**: Assess fluency, relevance, and completeness
5. **Interactive system**: Build web demo for public use

7.4 Impact

This work advances automatic lyrics understanding for Russian language, contributing to:

- Music information retrieval
- Russian NLP resources and benchmarks
- Explainable AI in cultural domains
- Practical applications for music education and appreciation

8 References

- Watanabe and Goto, 2020. *Lyrics Information Processing: Analysis, Generation, and Applications.* https://scholar.google.com/scholar?q=Lyrics+Information+Processing:+Analysis,+Generation,+and+Applications

- Sterckx et al., 2017. *Break it Down for Me: A Study in Automated Lyric Annotation.* https://arxiv.org/abs/1708.03492

- VUA Metaphor Detection shared tasks (NAACL 2018, ACL 2020). https://aclanthology.org/search/?q=VUA%20Metaphor%20Detection%20Shared%20Task

- Genius annotation platform. https://genius.com