

---

# Генерация поэзии с использованием методов машинного обучения

*Вамбриков Никита Б05-251*

---

# Проблемы генерации поэзии

- На данный момент генерация текстов уже хорошо решенная задача(в тч. и поэзии)
- Но даже DeepSeek не сразу справился с этой задачей
- Ты уходишь вдаль без оглядки (A)  
Я останусь здесь, в тишине (B)  
Но в душе ещё теплятся святки (B)  
Тех мгновений, что были вдвоём (A)
- Основная проблема при генерации это рифма и паттерн рифмы
- Также мы хотим запускать нашу модель локально
- Большинство методов в статьях используют генерацию тестов справа налево, что требует обучение модели с нуля

---

# Подход с генерацией последнего слова первым

- Мы хотим генерировать текст, в следующем формате:
- A:rain <sep> ... rain
- A:rain <sep> ... pain
- При этом наш паттерн рифмы был А А

---

# Что хотим от модели

- На вход хотим указывать метаданные песни(название, жанр, эмоцию и тд)
- <title>...<genre>... ...<rythm\_pattern>
- На выходе текст с генерацией последнего слова первым
- Запускалась локально
- Была рифма в соответствующем паттерне

---

# Предобработка текстов

- Я использовал датасет собранный с <https://genius>, сайт с текстами песен и метаданными
- Для создания паттерна используем последние слова песни и проверяем рифму через <https://github.com/bootphon/phonemizer>

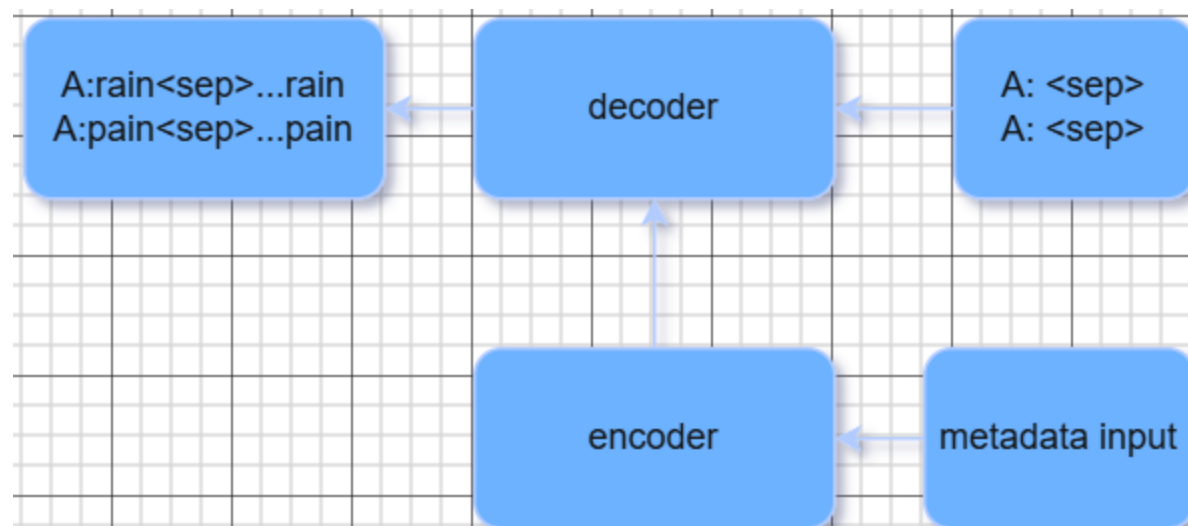
---

# Проблема обучения с нуля

- Обучение с нуля требует гораздо больше ресурсов, также модель перестает понимать смысл слов.
- Решение: использовать предобученную модель.
- Плюсы: модель понимает язык и быстрее обучается

# Модель

- Благодаря encoder-decoder архитектуре нам подойдет модель T-5
- Можем сразу добавлять ритм токены в декодер



---

# Гипотезы

- Ожидается что обученная модель будет лучше моделей генерирующих слева направо, т.к мы используем pretrain модель
- Будет в первую очередь решен вопрос рифмы и при этом текст будет иметь смысл



# Метрики

- С помощью той же библиотеки мы можем считать следующие метрики
- Они должны быть значительно лучше в сравнении с pretrain моделью
- Также хотим метрики разнообразия для более интересных и разнообразных тестов

$$P = \frac{1}{|R|} \sum_{t_i, t_j}^R \text{rhyme}(t_i, t_j)$$

$$FPR = \frac{1}{|NR|} \sum_{t_i, t_j}^{NR} 1 - \text{rhyme}(t_i, t_j)$$

$$\text{distinct-}n = \frac{\# \text{уникальных } n\text{-грамм}}{\# \text{всех } n\text{-грамм}}.$$

# Результаты

RP, %	RFP, %	distinct-4, %
52.5	<b>6.9</b>	60.5
<b>94.1</b>	11.3	<b>87.8</b>

По итогам предоставлено две две модели.

Первая обученная с нуля, вторая предобученная.

Как мы видим по метрикам вторая модель лучше почти во всем

Из чего следует что, наш способ лучше чем генерация слева направо требующая полного переобучения

---

# Примеры генерации

- Первая модель
- RHYME\_A now<SEP>Look at me now<sentence\_end>
- RHYME\_B you<SEP>I'm so in love with you<sentence\_end>
- RHYME\_C do<SEP>And I don't know what to do<sentence\_end>
- RHYME\_D me<SEP>But you've got a hold on me<sentence\_end>
- RHYME\_D see<SEP>You're the only one I see<sentence\_end>
- RHYME\_D be<SEP>That I wanna be<sentence\_end>

# Вторая модель

RHYME\_A now<SEP>Look at me now<sentence\_end>  
RHYME\_B you<SEP>I'm so in love with you<sentence\_end>  
RHYME\_C do<SEP>And I don't know what to do<sentence\_end>  
RHYME\_D me<SEP>But you've got a hold on me<sentence\_end>  
RHYME\_D see<SEP>You're the only one I see<sentence\_end>  
RHYME\_D be<SEP>That I wanna be<sentence\_end>

Промт моделей:

<title>Look at Me Now<artist>Charlie Puth<schema>RHYME\_A RHYME\_B RHYME\_C  
RHYME\_D RHYME\_D RHYME\_D</s>

---

# Конец

Спасибо за внимание