Классификация русских классических стихов по эпохам

Пономарев Андрей Сергеевич

2024–2025 учебный год,
весенний семестр

Аннотация

Настоящая работа посвящена применению современных средств обработки естественного языка для анализа русских стихотворных текстов. Решается задача классификации русской классической поэзии по эпохам. Для задачи составлен и размечен датасет. Обучены две модели: LSTM-нейросеть, использованная с готовыми эмбеддингами, а также классификатор на основе кодировщика RuBERT-tiny2. Второй показывает точность около 70%.

Репозиторий проекта находится по ссылке
`https://github.com/ponomarevandr/russian_poems_classification`.

1 Введение

Обработка естественного языка (natural language processing) – область информатики, в настоящее время переживающая бурное развитие. Ее методы находят широкое применение в самых разных прикладных задачах. Из-за подобной широты многие задачи все еще далеки от исчерпывающего анализа, существующие решения можно свободно дорабатывать и развивать.

Одним из разделов обработки естественного языка, которые поныне не находятся в центре внимания, является анализ стихотворных текстов. Об этом свидетельствует как относительно малое число публикаций по теме, так и слова их авторов напрямую (например, [BGC21]).

В настоящем проекте предлагается решение задачи классификации русских классических стихотворений по эпохе, или, иначе, течению, направлению. Автор не нашел в Интернете статей, освещавших бы в точности этот вопрос, и в этом смысле работа нова.

Будут рассмотрены два подхода к решению задачи: с помощью нейронной сети LSTM (в качестве базового) и с помощью предобученной модели RuBERT-tiny2.

Лучший результат показала модель RuBERT-tiny2 – точность около 70%. Она в дальнейшем может применяться на практике, например, для категоризации стихотворений в Интернете, как классических, так и современных.

1.1 Команда

Проект подготовил **Пономарев А. С.**

2 Смежные работы

Перечислим несколько статей, близких проекту по тематике.

Работа [NMA12] посвящена классификации малайской поэзии по нескольким темам. Статья написана давно и использует ныне устаревший подход – метод опорных векторов (SVM) поверх TF-IDF, – но подходит для знакомства с задачей.

Интересна и обширна статья [Rum+22]. Авторы рассматривают задачу классификации стихотворений Хафиза Ширази (на персидском языке) по периодам в жизни автора, что совсем близко к теме настоящего проекта. Основная используемая модель – LSTM-нейросеть, но помимо нее авторы пробуют логистическую регрессию, SVM, random forest, Bi-LSTM, GRU; в качестве эмбеддингов – CBOW, SkipGram, а также их конкатенацию. Авторы утверждают, что добились внушительной точности в 85%.

Отметим также близкую по тематике работу [ORE20], посвященную классификации арабской поэзии по эпохе написания. Авторы используют CNN-нейросеть с эмбеддингами FastText, относительно большой датасет (десятки тысяч стихотворений) и получают точность порядка 80%.

В работе [BGC21] исследуется применение моделей-трансформеров для разбиения испанских сонетов по категориям выражаемых чувств. Для различных классов авторы добиваются F1-метрики от 0.6 до 0.8.

В отдельную ветвь можно выделить исследования, приспосабливающие для анализа стихотворных текстов предобученные BERT-подобные модели. В работе [SRZ23] дообучена модель AraBERT для классификации арабской поэзии по выражаемым эмоциям (точность 76.5%). Статья [Ros+23] описывает масштабный труд по обучению модели ALBERTI – мультиязыкового кодировщика для анализа стихотворений – на огромном массиве данных в миллионы произведений. Авторы утверждают, что на момент создания как сама модель, так и показанные результаты не имели аналогов. В работе также затронута интересная тема анализа средствами машинного обучения метрики стиха (то есть рисунка слогов и ударений).

3 Данные

Для работы с моделями был создан и размечен датасет русской классической поэзии. За основу взяты два ранее существовавших датасета стихов, размещенные в открытом доступе: [GG22] и [Sil20].

После объединения исходных датасетов данные были приведены к единому формату и очищены (см. Юпитер-тетрадку *dataset_preparation.ipynb*). Итоговая структура датасета показана в таблице 1.

Поле *author* всюду приведено к одной лишь фамилии автора; в случае совпадений используются приписки через дефис, например: *Багрицкий-отец*, *Иванов-В*. Исключены иноязычные авторы, представленные в переводах.

Заглавие в поле *title* приведено к нижнему регистру и очищено ото всех небуквенных символов, кроме цифр и пробелов.

В текстах стихотворений по возможности исправлены ошибочные символы, удалены теги, сноски, номера строф, символьные артефакты. Убраны пустые строки. Исключены записи, состоящие преимущественно не из русских букв или имеющие среднюю длину строки более 60 символов.

Из датасета удалены дубликаты. Их поиск велся как по заглавию, так и по расстоянию Левенштейна между текстами.

	author	**epoch**	**title**	**part**	**text**
Содержание	Фамилия автора	Эпоха (течение, направление)	Заглавие	Номер фрагмента	Текст фрагмента
Формат	Русские буквы: заглавная, далее строчные; дефис	Одно из: классицизм, золотой век, критический реализм, серебряный век, футуризм, соцреализм, шестидесятники	Строчные русские и английские буквы, цифры, пробел	Индекс с нуля	8–40 строк, без пустых строк

Таблица 1. Структура датасета.

При работе с моделью было решено использовать стихотворные тексты длиной от 8 до 40 строк включительно. Для этого все более длинные произведения были «нарезаны» на фрагменты допустимой случайной длины. Эмпирически установлено, что распределение длин коротких стихотворений в строках близко к гамма-распределению, $\xi \sim \mathrm{Gamma}(\alpha, \beta)$, с матожиданием $\mathrm{E}\xi \approx 18.23$ и дисперсией $\mathrm{D}\xi \approx 81.66$. Напомним, что его плотность определяется формулой

$$p_{\alpha,\beta}(x) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)},$$

а также верно $\mathrm{E}\xi = \alpha/\beta$ и $\mathrm{D}\xi = \alpha/\beta^2$. Случайные длины при разбиении стихотворений на фрагменты получались округлением элементов выборки из описанного распределения.

Поле *epoch* является целевым признаком. Будем называть его эпохой стихотворения (синонимы: течение, направление). Выделим семь основных эпох: см. таблицу 2. Каждая эпоха имеет своеобразные черты и примерную датировку, представление о которых дают многочисленные открытые источники. Однако автору неизвестно единого исследования, которое давало бы исчерпывающее и достаточно строгое разбиение русской поэзии по эпохам, поэтому он в немалой степени руководствовался личной интуицией и чувством прекрасного.

Для простоты будем считать, что эпоха стихотворения определяется эпохой его автора; 189 авторам, представленным в датасете, были присвоены метки в соответствии со сказанным выше, а затем перенесены на соответствующие фрагменты произведений. Списки авторов по эпохам см. в Юпитер-тетрадке *dataset_ labeling.ipynb*.

Получившееся разбиение на классы относительно сбалансированно: самый большой – Серебряный век – содержит 16204 стихотворных фрагмента; самый малый – шестидесятники – 3238, что примерно в 5 раз меньше, и сразу за ним – классицизм – 3281.

Для работы с моделями датасет разбивался на три части: тренировочную (train), валидационную (eval) и тестовую (test) – в соотношении 70 : 15 : 15.

Эпоха	Примерная датировка	Комментарий
Классицизм	XVIII век	
Золотой век русской поэзии	Первая половина XIX века	
Критический реализм	Вторая половина XIX века	Близко к понятию «натуральной школы»
Серебряный век русской поэзии	Начало XX века	Близко к понятию символизма
Футуризм	Начало XX века	Может рассматриваться как часть Серебряного века, но резко выделяется и достаточно объемен
Социалистический реализм	Середина XX века	
Поэзия шестидесятников	Вторая половина XX века	

Таблица 2. Эпохи, выделяемые в русской поэзии.

4 Модели

4.1 LSTM

Первая использованная модель, и она же взятая в качестве базового решения (baseline), – это нейронная сеть типа Long short-term memory (LSTM) (см. Юпитер-тетрадку *training_ lstm.ipynb*). Для работы выбрана известная реализация из библиотеки Pytorch – класс *nn.LSTM*. Поверх стандартной LSTM-нейросети добавлен полносвязный линейный слой (*nn.Linear*) с функцией активации LogSoftmax для получения логарифмированных вероятностей классов.

Для кодирования текста используются готовые русскоязычные эмбеддинги *word2vec-ruscorpora-300* из библиотеки Gensim. Для их работы требуется лемматизация и определение части речи; то и другое производится средствами библиотеки Pymorphy3 пословно.

Нейросеть *nn.LSTM* имеет размерность входа 300, размерность внутреннего состояния 400 и единственный скрытый слой.

Для обучения использовался оптимизатор *optim.Adam* с параметром $lr = 1.5 \cdot 10^{-4}$ (learning rate).

4.2 RuBERT-tiny2

Вторая использованная модель – это BERT-подобный кодировщик RuBERT-tiny2 (см. Юпитер-тетрадку *training_ rubert.ipynb*), опубликованный в Интернете [Coi23]. Модель была выбрана, во-первых, потому, что специально предназначена для русскоязычных текстов, во-вторых, из-за своего сравнительно малого (как и обещает название) размера в 29.4 миллиона параметров.

Работа с моделью происходит через библиотеки Datasets, Evaluate и Transformers компании Hugging Face. Автор значительно опирался на код из Юпитер-тетрадки [Dal23]. Хотя исходно модель RuBERT-tiny2 – это кодировщик, а не классификатор, средства биб-

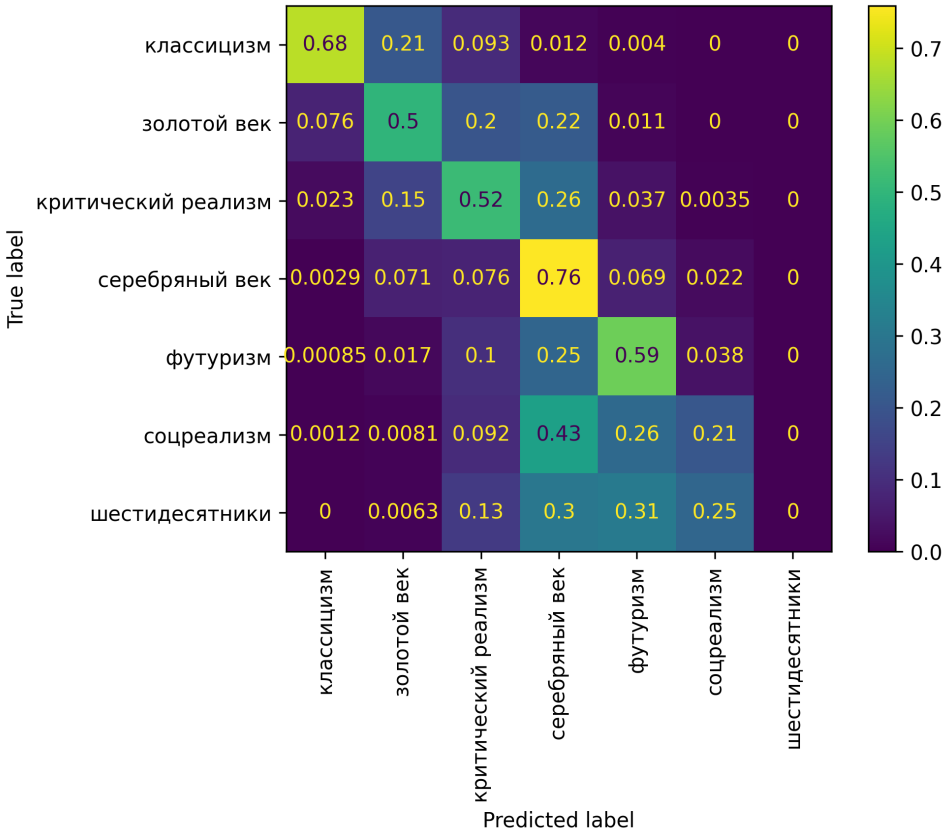

Рис. 1. Матрица ошибок LSTM-модели.

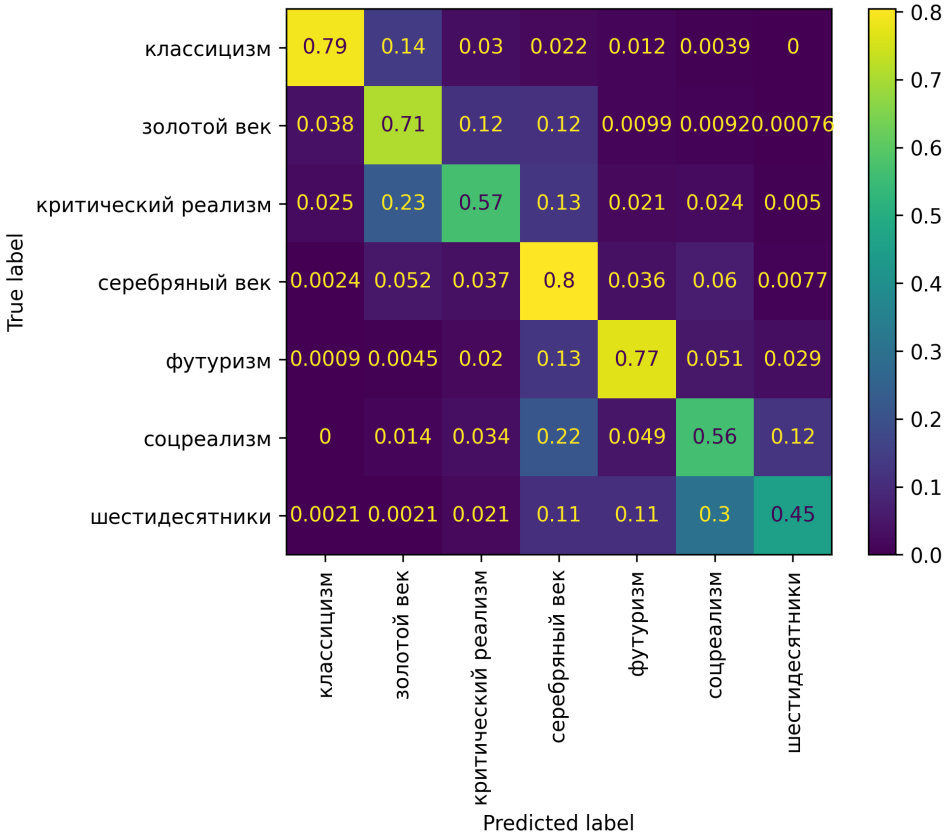

Рис. 2. Матрица ошибок модели RuBERT-tiny2.

Модель	accuracy	weighted F1
LSTM-нейросеть	0.543	0.514
RuBERT-tiny2	0.696	0.694

Таблица 3. Результаты моделей на тестовых данных.

лиотек позволяют загрузить ее в класс *AutoModelForSequenceClassification*, автоматически добавляя необходимые элементы. Для обучения использовался оптимизатор *optim.Adam* из библиотеки Pytorch, с параметром $lr = 10^{-5}$. Модель поддерживает работу на GPU (Cuda), что значительно ускорило обучение – средствами Google Colab оно было завершено приблизительно за 25 минут.

5 Результаты

Предсказания моделей оценены по двум стандартным метрикам: accuracy (точности) и weighted F1 – результаты в таблице 3.

Для полученных на тестовых данных предсказаний также построены матрицы ошибок (confusion matrices), см. рис. 1, 2.

LSTM-модель показала удовлетворительный результат. Она справляется с задачей примерно в половине случаев; притом никогда не выдает в качестве результата поэзию шестидесятников.

Дообученная модель RuBERT-tiny2 справляется с задачей хорошо, демонстрируя значительную точность примерно в 70%. Наиболее трудны для определения оказываются опять же шестидесятники, за ними – соцреализм и критический реализм. Легче всего оказывается распознать поэзию Серебряного века и классицизма. Заметим, что трудность определения обусловлена вовсе не только размерами классов.

6 Файлы моделей и демонстрация работы

Файлы обеих обученных моделей сохранены в репозитории. В Юпитер-тетрадке *training_rubert.ipynb* приведен минимальный код, открывающий файлы готовой модели и позволяющий запустить ее на любом тексте по желанию пользователя.

7 Заключение

- Составлен датасет русских классических стихотворений. Датасет размечен: добавлен целевой признак эпохи (течения, направления), всего 7 классов.
- Для решения задачи классификации обучена LSTM-нейросеть, использовавшаяся вместе с готовыми русскоязычными эмбеддингами. Модель показала удовлетворительный результат: точность порядка 50%.
- Для решения задачи классификации применена предобученная модель RuBERT-tiny2, основные достоинства которой – приспособленность к русскому языку и сравнительно малый размер. Дообученная модель показала хороший результат на тестовых данных: точность около 70%.
- Полученная на основе RuBERT-tiny2 модель может в дальнейшем использоваться на практике, например, для категоризации стихов на интернет-сайтах.

References

[BGC21] A. Barbado González, M. D. González Barbado, and D. Carrera. *Lexico-semantic and affective modelling of Spanish poetry: A semi-supervised learning approach*. Sept. 2021.

[NMA12] J. Noraini, M. Masnizah, and N. Sh. Azman. "Poetry Classification Using Support Vector Machines". In: *Journal of Computer Science* 8 (2012), pp. 1441–1446.

[Rum+22] J. F. Ruma et al. "A deep learning classification model for Persian Hafez poetry based on the poet's era". In: *Decision Analytics Journal* 4 (2022), p. 100111.

[ORE20] M. Orabi, H. El Rifai, and A. Elnagar. "Classical Arabic Poetry: Classification based on Era". In: *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*. 2020, pp. 1–6.

[SRZ23] S. Shahriar, N. Al Roken, and I. Zualkernan. "Classification of Arabic Poetry Emotions Using Deep Learning". In: *Computers* 12.5 (2023).

[Ros+23] J. de la Rosa et al. *ALBERTI, a Multilingual Domain Specific Language Model for Poetry Analysis*. July 2023.

[GG22] Giothun and Ilya Gusev. *Russian poetry corpus*. 2022. URL: `https://www.kaggle.com/datasets/greencools/russianpoetry`.

[Sil20] Georgy Silkin. *19000 russian poems*. 2020. URL: `https://www.kaggle.com/datasets/grafstor/19-000-russian-poems`.

[Coi23] David Dale (Cointegrated). *RuBERT-tiny2*. 2023. URL: `https://huggingface.co/cointegrated/rubert-tiny2`.

[Dal23] David Dale. *train-rubert-tiny-sentiment-classifier.ipynb*. 2023. URL: `https://gist.github.com/avidale/e678c5478086c1d1adc52a85cb2b93e6`.