ИНСТРУМЕНТЫ

Борис Орехов

ОТКРЫТЫЕ КОМПЬЮТЕРНЫЕ ИНСТРУМЕНТЫ ДЛЯ РЕШЕНИЯ ЗАДАЧ ОЦИФРОВКИ И АНАЛИЗА РУССКОЯЗЫЧНОГО ТЕКСТА В ОБЛАСТИ DIGITAL HUMANITIES

В статье дается обзор не очень известных модулей, которые можно использовать для решения задач Digital Humanities, связанных с текстовым анализом и оцифровкой. К таким модулям отнесены те, которые облегчают оцифровку текстов, напечатанных в дореформенной орфографии (OCR-модель и конвертер в новую орфографию), акцентуатор, расставляющий ударения, детектор прямой речи, код, позволяющий оценить формульность фольклорного текста, конвертер для формата TEI. В этом же ряду представлен модуль, облегчающий обработку текста для публикации в настоящем журнале.

Ключевые слова: python, digital humanities, old orthography, converter

Борис Валерьевич Орехов
НИУ ВШЭ, ИРЛИ РАН
nevmenandr@gmail.com

DOI: 10.31860/cgi-2025-2-71-83

Манифест

Даже[1] в эпоху почти всемогущих больших языковых моделей продолжают быть востребованы компьютерные инструменты, позволяющие решать частные задачи, потому что именно с таким классом задач LLM справляются хуже. Пока что современным версиям искусственного интеллекта недоступна тонкая нюансировка, завязанная на особенностях предметного поля.

Одновременно с этим информация о таких инструментах плохо распространяется и внутри сообщества, и среди внешних специалистов, потенциально заинтересованных во внедрении этих инструментов в свои продукты. Автору этих строк известно как минимум о двух попытках самостоятельно создать программный комплекс, дублирующий функционал одного из описанных ниже модулей для Python, несмотря на то, что модуль существует давно, показывает высокое качество работы и распространяется свободно. Проблема в информированности. Что тоже удивляет, учитывая, как много усилий вкладывается разработчиками поисковых систем в свои технологии, а через поисковые системы установить наличие обозреваемых ниже инструментов совсем не трудно. Возможно, тем, кто испытывает потребность в этих инструментах, кажется неправдоподобной мысль, что кто-то эту потребность уже удовлетворил, — настолько частной представляется задача, которую требуется решить — что они даже не пользуются поисковиками, чтобы на деле убедиться в том, в чем и так уверены.

Тем не менее, ситуация не остается статичной. В сфере Digital Humanities, в том числе и в России, постепенно происходит накопление полезной кодовой базы, и важной задачей сообщества становится информирование и самоинформирование о технических возможностях, которыми прирастает поле благодаря этой базе.

Окончательно решить проблему информированности не удастся: слишком насыщенным представляется современное информационное поле, а ресурсы сообщества на этом поле ограничены. Это, однако, не означает, что от решения поставленных задач следует отказаться. Мы с коллегами видим этот журнал как способ информирования всех заинтересованных, а этот текст — один из первых шагов в этом направлении.

Далее речь пойдет о разработанных отечественными компьютерными лингвистами и специалистами в области цифровых гуманитарных исследованиях инструментах, которые могут оказаться полезными для аналитической работы с текстами на русском языке.

Оговорка состоит в том, что в обзоре будут не все поддерживающие русский язык программные библиотеки, а только те, информация о которых нуждается в распространении. Во-первых, модули типа *natasha, deeppavlov* или *spacy*, с помощью которых также можно решать задачи анализа русскоязычного текста, и так достаточно известны, и даже входят в материалы соответствующих учебных курсов; узнать о них не представляет сложности. Информационной поддержки же требуют менее масштабные проекты, спектр возможностей которых не так широк, но инфраструктурная роль которых может оказаться важной в рамках предметного поля. Во-вторых, вышеназванные и другие подобные им библиотеки, предоставляющие функциональность для анализа текста, в достаточно сильной степени практически ориентированы на обеспечение потребностей компьютерной лингвистики, смежной, но не тождественной цифровым гуманитарным исследованиям области. В настоящем тексте я сосредоточу внимание именно на DH-ориентированных модулях, которые могут оказаться полезными и компьютерным лингвистам, но по остаточному принципу.

Все модули, упомянутые в этом обзоре, представляют собой открытое, свободно и бесплатно распространяемое программное обеспечение, функционирующее как расширение для базовой сборки интерпретатора языка программирования Python. Все они были созданы либо под руководством либо усилиями преподавателей школы лингвистики московского кампуса НИУ ВШЭ.

Обозреваемые модули удобно разделить на три группы, основываясь на сфере их применения. В первой группе оказываются модули, рассчитанные на работу с русскоязычными текстами в старой (дореформенной) орфографии. Во второй — модули, помогающие в работе с преимущественно художественными текстами или текстами, которые ставятся в один ряд с художественными. В третьей группе, состоящей из одного модуля, находится код, помогающий в подготовке публикаций в настоящем журнале.

Старая орфография

Реформа русской орфографии произошла в 1917 году, разделив историю отечественного письма на «до» и «после». Из текстов исчезли некоторые буквы, отменено обязательное добавление буквы «ъ» к слову, оканчивающемуся на согласный, изменены правила написания некоторых приставок. Помимо всех культурных и семиотических проекций этого шага, нужно учитывать и технические.

Для русского языка существует достаточно мощное оснащение инструментами автоматической обработки текста, включающими весь основной функционал такого рода — от морфологического анализа до извлечения именованных сущностей. Но все эти языкозависимые инструменты адаптированы для анализа текста в новой, пореформенной орфографии. Таким образом, дореволюционные тексты оказываются без технического обеспечения. При этом специалистам по цифровым гуманитарным исследованиям часто приходится иметь дело с культурно значимыми текстами, созданными в дореформенную эпоху и с тех пор не переиздававшимися. Уже сам процесс их оцифровки связан с известными сложностями, поскольку системы OCR для русского языка настроены на распознавание современных текстов. Такие инструменты не справляются ни с отдельными символами, свойственными старой, но не новой орфографии, ни с целыми словами, изменившими свой облик после реформы.

OCR-модель для распознавания текстов в старой орфографии

Созданию модели предшествовала публикация датасета распознанных и вручную исправленных текстов в старой орфографии [Orekhov 2024a]. На этих данных была обучена модель, использующая инфраструктуру tesseract-ocr.

Ссылка: https://huggingface.co/Serovvans/trocr-prereform-orthography

Достоинствами инструмента являются легкость установки и использования. Зависимости инструмента: *tesseract-ocr* и *poppler* и пакеты для Python: *pytesseract* (версии 0.3.10 и выше), *Pillow* (10.0.0), *transformers* (4.46.0), *torch* (2.5.0), *numpy* (1.23.0), *tqdm*, *pdf2image* (1.17.0, для распознавания книги целиком). Все они устанавливаются при помощи пакетных менеджеров в тех системах, где такие менеджеры доступны.

Код, вызывающий соответствующие функции, также не переусложнен:

```
from recognize_page import recognize_page
page_path = "page.png"
text = recognize_page(page_path,
text_output_path="output/file.txt")
print(f"Текст страницы:\n")
```

```
print(text)
```

Таким образом можно вывести на экран текст, размещенный на одной распознанной странице. Есть отдельная функциональность для распознавания текста целиком.

Качество распознавания, которое предоставляет модель, можно охарактеризовать как среднее:

- CER (Char Error Rate) = 0.095
- WER (Word Error Rate) = 0.298

Эти показатели означают, что если отдельные символы распознаются системой довольно уверенно, то с целыми словами возникают сложности, приводящие к тому, что правильно прочитывается только около 70 % лексических единиц. Таким образом, после применения этого инструмента требуется дополнительный контроль качества распознавания и корректировка результатов, с чем как раз довольно неплохо справляется большая языковая модель.

Вот результаты эксперимента с GPT-4o (лето 2025 года):

Запрос: Ниже приведен текст на русском языке в дореформенной орфографии, в нем содержатся ошибки распознавания после OCR. Исправь ошибки, сохрани старую орфографию.

дат-

скій король Христіанъ VII посѣтилъ академію паукъ.

Ответ: Ваш текст, исправленный с сохранением дореформенной орфографии:

датскій король Христіанъ VII посѣтилъ академію наукъ.

Ошибки распознавания исправлены, а орфография текста соответствует дореформенному порядку.

Еще одно уточнение следует ввести при разговоре о русскоязычной типографике XVIII века. Известно, что тексты этого времени отличаются от позднейших в том числе формой некоторых букв. Например, буква «т», приобретшая современный облик еще в XIX веке, за 100 лет до того напоминала скорее букву «m». Для таких текстов обозреваемая модель подходит плохо, нужно использовать альтернативные решения, некоторые из них разработаны на базе программы Transkribus, но существуют только в ее экосистеме. Перспективно выглядят модели «Russian print of the 18 c. (V. Okorokov's

Printing House)»[2] и «Russian print XVIII cent PyLaia»[3]. При их использовании, однако, приходится учитывать, что и сама программа Transkribus, и созданные для нее модели не являются свободным и бесплатно распространяемым продуктом, а значит, вся эта экосистема экзистенциально связана со стоящей за ней частной предпринимательской структурой, перспективы существования которой не всегда ясны.

Возвращаясь к основной обозреваемой разработке, следует добавить, что в ее описании присутствует абсурдная помета: «The model is trained to recognize printed texts in Old Russian language». Поскольку в основу модели легли данные текстов на современном русском, а не древнерусском языке, можно с уверенностью сказать, что этот тезис не соответствует действительности. Очевидно, создатели модели перепутали древнерусский язык и современный русский язык, фиксируемый на письме при помощи правил старой орфографии. У этих сущностей действительно есть некоторое внешнее сходство. Например, частично пересекается набор графем (буквы «ять», «фита», «ижица», «и десятеричное»), отсутствующих в современной системе письма. Однако даже сам набор графем в дореформенной орфографии и древнерусской системе письма не тождествен. В старой орфографии отсутствуют древнерусские йотированные варианты букв, предназначенных для обозначения гласных фонем, юсы (большой и малый), написание под титлом, древнерусская пунктуация. Старая орфография наследует древнерусской письменности, но наследие не подразумевает тождества. Но главное — современный русский и древнерусский языки серьезным образом отличаются с точки зрения лингвистического описания. Они имеют разный лексический состав (среди бросающихся в глаза частотных лексико-грамматических отличий — в древнерусском языке иначе выглядят местоимения), разные грамматические системы (отличаются склонение имен, формы глагольного времени и т. д.). Отличить современный русский язык в старой орфографии и древнерусский язык с легкостью сможет любой носитель, даже не располагающий специальной лингвистической подготовкой.

Если не учитывать этого недостатка описания модели, она вполне пригодна к использованию как бесплатная альтернатива коммерческим решениям.

Модуль для транслитерации старой орфографии в новую

После того, как текст оцифрован, его можно обработать с помощью соответствующих программных средств. Но, как уже было сказано, большая часть из них языкозависимые и настроены на современное написание. Коммерческие компании, делающие существенный вклад в появление и распространение программных решений в области обработки текста, не заинтересованы в текстах в старой орфографии по очевидным причинам: не существует заказчиков, располагающих достаточным количеством подобных данных, и способных оплатить их обработку. Чтобы получить возможность беспрепятственного использования наличных инструментов, следует конвертировать текст, написанный в дореформенной орфографии, в новую орфографию. Для этого подходит модуль для Python *prereform2modern*.

Ссылка: https://pypi.org/project/prereform2modern/

Достоинства модуля в качественной работе и отсутствии сложных зависимостей, что облегчает установку. Все преобразования совершаются при помощи встроенных в базовую сборку Python средств работы со строками.

Несмотря на узкую специализацию модуля, он обладает достаточно широким спектром настроек. Тексты можно преобразовывать, сохраняя информацию о внесенных изменениях, а результат преобразования может быть представлен как в виде простого текста, так и в виде фрагмента XML, сфомированного по правилам TEI:

```
<choice>
<reg>пример</reg> <orig>примеръ</orig>
</choice>
```

Функциональность модуля можно вызывать и из командной строки, и из интерфейса интерпретатора Python. Модуль несколько лет не обновлялся, но это не характеризует его с худшей стороны, так как качество его работы с самого начала было на достойном уровне, и обновления коду не требуются. Здесь будет уместно привести бурлескное стихотворение, иллюстрирующее случай обозреваемого инструмента:

> бог создал труд и обезьяну
> чтоб получился человек
> а вот пингвина он не трогал
> тот сразу вышел хорошо

Художественные и парахудожественные тексты

Модуль для акцентуации русского поэтического текста

Расстановка ударений в русскоязычном тексте — действие, которое необходимо для многих типов аналитических операций, но главным образом они связаны с областью интересов стиховедения. Ударения могут быть точкой отсчета для автоматизированного определения силлаботонического размера, установления конфигурации ритма строки, метр которой уже определен, для нахождения ритмизованных отрезков прозы. Разумеется, границы применения такого инструмента шире дисциплинарных рамок цифровых гуманитарных исследований, и включают интересы специалистов, работающих с учебными материалами для иностранцев, с программным обеспечением для синтеза речи и т. д. Трудности, которые встают перед создателями такого инструмента, заключаются в том, что основанный на словарях и правилах, он не сможет покрыть все лексическое разнообразие русского языка, а выстроенный на базе машинного обучения будет допускать ошибки в простых однозначных и при этом частотных случаях. Купировать эти проблемы способен программный модуль, заимствующий сильные стороны обоих подходов. Он был создан в 2022 году [Короткова 2022].

Ссылка: https://pypi.org/project/ru-accent-poet/

Модуль решает задачу постановки ударения, используя встроенный словарь и подключение модели рекуррентной нейросети, где это необходимо. Такой подход сразу повышает качество результата, метрики приведены в статье [Короткова 2022].

Единственным минусом обозреваемого программного обеспечения является скорость работы. Чтобы можно было сразу воспользоваться размеченными текстами, не дожидаясь, пока отработает программа, в 2024 году был опубликован датасет текстов классической русской прозы, размеченной при помощи *ru-accent-poet*. [Orekhov 2024b], кроме того, на этих данных построены уже опубликованные исследования [Орехов 2022].

Модуль для определения прямой речи персонажей в художественном тексте

Зона прямой речи является значимой и маркированной зоной в тексте художественной прозы. Ей было посвящено несколько работ в течение прошлого десятилетия (например, [Хисамова 2013]), некоторые из них были количественными: [Sobchuk 2016], но от-

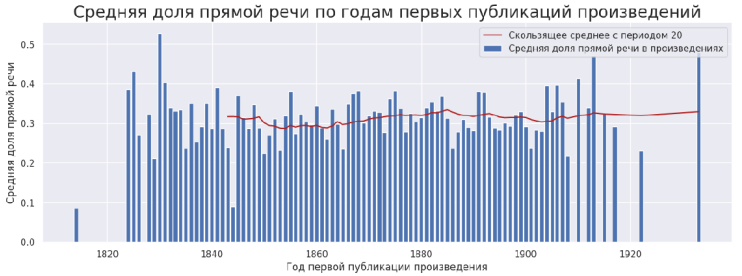

Рис. 1. Средняя доля прямой речи по годам первых публикаций произведений (рисунок автора модуля Д. Есаян)

сутствие возможности создавать прозрачные и репрезентативные выборки ограничивало свободу маневра исследователей. Обозреваемый модуль призван решить эти проблемы, одновременно сформулировав пользу для другого рода операций, связанных, например, с областью выравнивания двуязычных текстов для параллельных корпусов.

Ссылка: https://pypi.org/project/direct-speech-extractor-ru/

У модуля высокие значения метрик оценки: accuracy = 0.989, precision = 0.998, recall = 0.96 и f-measure = 0.979. При этом повторенное с помощью этого инструмента исследование [Sobchuk 2016] не подтвердило его выводов: на протяжении всего XIX века реплики литературных персонажей русской прозы составляли приблизительно одинаковую долю в тексте объемом в 30 % (см. рис. 1) в то время как О. Собчук пишет о ярко выраженном скачке этой доли.

Таким образом, можно констатировать, что сферу цифровых гуманитарных исследований догнал кризис воспроизводимости [Pashler, Wagenmakers 2012].

Модуль для оценки формульности фольклорного текста

Ссылка: https://pypi.org/project/formularity-rfs/

Формульность (повторяемость, стереотипность) текста в фольклоре — одна из системообразующих характеристик, она определяет облик текста, облегчает его устное бытование и передачу от одного носителя другому. Цифровой гуманитарный подход подразумевает

возможность перевести это понятие в термины исчисляемых категорий, что открыло бы возможность для сравнения разных текстов по параметру формульности.

Функциональность обозреваемого модуля построена на авторском алгоритме количественной оценки формульности текста, включающем разнообразные текстовые метрики: коэффициент вариативности словарного запаса (VocD), коэффициенты n-грамм, биномов, фразеологизмов и междометий.

Так выглядит размеченный программой текст (полужирным обозначены формульные в рамках анализируемого текста n-граммы и фразеологизмы из встроенного в модуль списка):

> Под ракитою зеленой Русский раненый лежал, Ко груди, штыком пронзенной, Крест свой медный прижимал. Кровь лилась из свежей раны На истоптанный песок; Над ним вился **черный ворон**, Чуя лакомый кусок. «Ты не вейся, **черный ворон**, Над моею головой! Ты добычи не дождешься, Я солдат еще живой! Ты слетай в страну родную, Отнеси маменьке поклон. Передай платок кровавый Моей женке молодой. Ты скажи: она свободна, Я женился на другой. Я нашел себе невесту В **чистом поле**, под кустом; Моя сваха – вострa сабля, И венчал граненый штык; Взял невесту тиху, скромну И приданно небольшо. Взял приданно небольшое – Много лесу и долин, Много сосен, много елок, Много, много вересин»
>
> **Найденные n-граммы:**
>
> • черный ворон
>
> **Коэффициент формульности:** 1.207

В таблице 1 даны агрегированные результаты анализа нескольких текстовых произведений.

Название песни	Биномы	n-граммы	VocD	Итог
У зари-то, у зореньки	0	0,1	0,786	0,886
По улице мостовой	0,1	0,3	0,862	1,262
Выйду ль я на реченьку	0,1	0,6	0,844	1,544
Вниз по матушке по Волге	0	1,3	0,688	1,988
Не одна во поле дороженька	0	0	0,643	0,643

Название песни	Биномы	n-граммы	VocD	Итог

Таблица 1. Результаты анализа песен из сборника «40 русских народных песен» (1988), сост. Юрий Зацарный (результаты получены автором модуля В. Сидненко)

Модуль зависит от нескольких сторонних пакетов: *spacy, pymorphy2, numpy, sklearn*, что грозит потенциальными проблемами с установкой и обновлениями, но так как речь идет о достаточно известных и широко распространенных пакетах, есть надежда, что сообщество поможет с решением.

Модуль для конвертации текстов в формате TEI

Ссылка: https://pypi.org/project/TEItransformer/

TEI на основе XML — это конвенциональный формат хранения оцифрованных текстовых данных. С самого начала его появления подразумевалось, что сохраненные таким образом произведения будет легко конвертировать в любой другой необходимый формат, предназначенный для обработки или публикации. На деле универсальных решений скроссформатной конвертации долгое время не существовало. Модуль *TEItransformer* призван заполнить эту лакуну [Kostyanitsyna, Skorinkin 2024]. Было бы удобно, если бы он был построен на чистом Python или хотя бы использовал в качестве зависимостей только модули Python, но в реальности создать такой конвертер оказалось слишком сложно, поэтому *TEItransformer* представляет собой, по сути, обертку вокруг XSLT-преобразований, устаревающей и ненадежной технологии обработки XML.

Модуль позволяет конвертировать исходный TEI XML в docx (универсальный редактируемый формат), html (удобный для веб-публикации), json (универсальный формат для анализа данных). Для этого используются пользовательские сценарии, которые описывают, как именно следует преобразовывать исходный текст. При этом текст, преобразованный в html, включает не только статическое представление строк, но и инструменты поиска по разным полям исходной разметки. На рис. 2 показано, как можно искать все реплики персонажа Маша на веб-странице, сгенерированной из TEI-представления драматического текста.

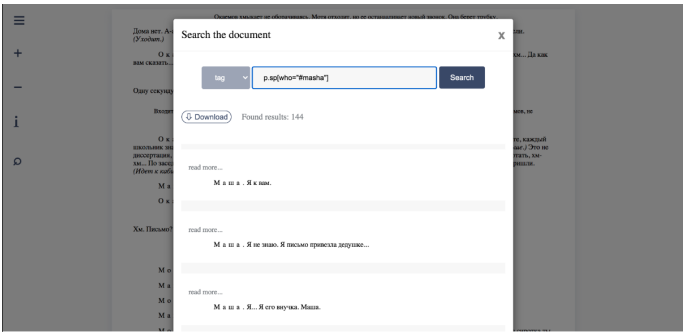

Рис. 2. Поисковый интерфейс на веб-странице, сгенерированной модулем *TEItransformer*

«Цифровые гуманитарные исследования»

Модуль предобработки текстовых материалов для журнала «Цифровые гуманитарные исследования»

Ссылка: https://pypi.org/project/cgi-processor/

Настоящий журнал готовится с помощью издательской системы Latex. Помимо очевидных редакторских действий подготовка включает и ряд технических преобразований, подразумевающих соответствие правилам отечественной типографики. Эти правила включают неразрывные пробелы определенной длины между фамилией и инициалами, короткое тире в числовых диапазонах и т. д. В Latex эти условия задаются при помощи специальных символов. К примеру, короткое тире в исходном коде Latex выглядит как два дефиса подряд. Ручная обработка текста, имеющая целью расстановку в тексте всех специальных символов, представляет собой трудоемкий процесс, в котором неизбежны множественные ошибки. Эту проблему призван решить модуль *cgi-processor* [Orekhov 2025]. В первую очередь он обрабатывает те случаи, которые специально учтены в системе стилей журнала «Цифровые гуманитарные исследования», хотя, вероятно, будет полезен и другим авторам и редакторам, работающим с русскоязычными текстами в Latex.

Обозреваемый модуль облегчает и ускоряет подготовку номера журнала, работающего на инфраструктурное единство и информационную проницаемость русскоязычного сообщества Digital Humanities.

Примечания

[1] Исследование осуществлено в рамках Программы фундаментальных исследований НИУ ВШЭ

[2] https://app.transkribus.org/models/public/text/44358

[3] https://app.transkribus.org/models/public/text/russian-print-xviii-cent

Литература

Исследования

Короткова 2022 — Короткова Ю. О. Комбинированный словарно-нейросетевой акцентуатор для разметки русского поэтического текста // Труды Института русского языка им. В. В. Виноградова. 2022. № 3 (33), С. 181–190. DOI: 10.31912/pvrli-2022.3.11

Орехов 2022 — Орехов Б. В. Случайные метры в русской прозе XIX века // Вещество поэзии: К 70-летию Юрия Борисовича Орлицкого: Сборник научных статей. М.: РГГУ, 2022. С. 24–30.

Хисамова 2013 — Хисамова Г. Г. Функции диалога в художественном тексте // Вестник Нижегородского университета им. Н. И. Лобачевского. 2013. № 6 (2). С. 245–247.

Kostyanitsyna, Skorinkin 2024 — Kostyanitsyna A., Skorinkin D. A Python Library for TEI Conversion into Edition Formats // Texts, languages and communities: 24th Annual Conference and Members' Meeting of the Text Encoding Initiative Consortium. Ciudad Autónoma de Buenos Aires: Universidad del Salvador, 2024. P. 114–115.

Orekhov 2025 — Orekhov B. Модуль предобработки текстовых материалов для журнала «Цифровые гуманитарные исследования» [Computer software]. 2025 URL: https://github.com/nevmenandr/cgi_processor

Orekhov 2024a — Orekhov B. Russian-Old-Orthography-Ocr (Revision 6f60636) // Hugging Face, 2024. DOI: 10.57967/hf/3280

Orekhov 2024b — Orekhov B. accentual-syllabic-verse-in-russian-prose (Revision 489fea1). // Hugging Face, 2024. DOI: 10.57967/hf/2438

Pashler, Wagenmakers 2012 — Pashler H., Wagenmakers E. Editors' Introduction to the Special Section on Replicability in Psychological Science // Perspectives on Psychological Science. 2012. N 7. P. 528–530.

Sobchuk 2016 — Sobchuk O. The Evolution of Dialogues: A Quantitative Study of Russian Novels (1830–1900) // Poetics Today. 2016. N 37 (1). P. 137–154.