# `lingcorpora`: СОЗДАНИЕ API ДЛЯ ЯЗЫКОВЫХ КОРПУСОВ

Кошевой А.Г. (alexeykochevoy@gmail.com)
НИУ ВШЭ, Школа Лингвистики, Москва, РФ

**Аннотация**

In this paper, I describe the results of the development of the tool for unified corpora access: `lingcorpora` Python package. The package provides access to different language corpora using uniform functions architecture. It saves users trouble dealing with limitation of the concrete corpora such as number of examples per page limitations, number of pages limitations, key word in context format support limitations, downloading results limitations and so on. Nowadays it provides access to the National Corpus of Polish Language, the National Corpus of Russian Language and Das Wortauskunftssystem zur deutschen Sprache in Geschichte und Gegenwart. But the quantity of accesible corpora is going to be improved over time. The package is publicly available and it is expected to evolve in the future.

**Keywords:** corpus linguistics, linguistic typology,

## 1 Введение

Многие лингвистические исследования базируются на корпусных данных, поэтому быстрый доступ к ним важен для исследователей. Например, при проведении типологических исследований может понадобиться обратиться сразу к нескольким корпусам с большим количеством запросов. Делать это вручную — длительный процесс.

Корпуса во многом различаются: в некоторых отсутствует возможность скачать выдачу, в других нет возможности ограничить количество примеров, иногда морфологические теги скрыты. Пакет `lingcorpora` предоставляет пользователям доступ к корпусам и позволяет добавлять и настраивать параметры выдачи, а также сохранять её в удобном для пользователя формате, не заходя на сайты корпусов. Полученные данные можно использовать для дальнейшей обработки данных путём подключения различных инструментов обработки естественного языка.

На данный момент существуют R-версия пакета (разрабатывается Г. А. Морозом) и Python-версия, проблемы разработки которой и описаны в данной работе.В Python-версии пакета реализован поиск по Национальному Корпусу Польского Языка (далее NKJP, см. Przepiórkowski et al. [2008], http://nkjp.pl), Национальному Корпусу Русского Языка (далее НКРЯ, см. Молдован [2007], http://www.ruscorpora.ru/) и Das Wortaus- kunftssystem zur deutschen Sprache in Geschichte und Gegenwart (DWDS, см. Klein and Geyken [2010], https://www.dwds.de). Также Катя Герасименко написала функцию для Center of Chinese Linguistics Corpus (CCLC, см. Huang et al. [2005], http://ccl.pku.edu.cn:8080/ccl_corpus/index.jspf), Георгий Алексеевич Мороз в R-версии пакета написал функции для аварского (http://baltoslav.eu/avar/index.php) и абхазского (http://baltoslav.eu/apsua/index.php) корпусов. Корпуса из R-версии будут в ближайшее время включены в Python-версию, планируется добавить и другие корпуса, например, RuTenTen, ГИКРЯ, Aranta, DeReKo и тд.

## 2 Описание пакета

Для достижения основной цели пакета — создание API для различных лингвистических корпусов — было решено создать несколько одинаковых функций, по функции на каждый корпус. В настоящий момент доступны функции `pol_search` — поиск в NKJP, `rus_search` — поиск в НКРЯ и `deu_search` — поиск в DWDS. Функция `pol_search` поддерживает все аргументы, `rus_search` — все, кроме аргумента `tag`, но в скором времени поддержка этого аргумента будет реализована. Названия функций устроены одинаково – в них фигурирует iso-код языка корпуса.

Для работы функций используется фиксированный набор аргументов для того, чтобы облегчить автоматизацию запросов. Всего используется шесть аргументов, одинаковых для R- и Python-версий:

- `query` – запрос, на вход принимается строка (допустимо использование регулярных выражений, если они разрешены корпусом)
- `corpus` – подкорпус, по умолчанию стоит основной корпус, на вход принимается строка
- `tag` – демонстрация морфологических тегов (пока что работает только в НКПЯ), на вход принимаются логические операторы True/False (по умолчанию False)
- `n_results` – количество результатов в выдаче, на вход принимается число (по умолчанию 10)

- `kwic` – режим `key word in context`, на вход принимаются логические операторы True/False (по умолчанию True)
- `write` – запись в `.csv`, на вход принимаются логические операторы True/False (по умолчанию False)

Как видно из списка выше, многие аргументы содержат значения по умолчанию. Это сделано для того, чтобы пользователь мог использовать только нужные в данный момент аргументы. К тому же, это существенно ускоряет работу самой функции. Возможно, при добавлении новых корпусов, список аргументов будет несколько меняться, чтобы можно было создать универсальный набор аргументов, который не отражал бы специфику конкретных корпусов.

Приведем пример запроса функции `pol_corpus()`:

```
lingcorpora.pol_search(query = 'tata', write = True)
```

Данный запрос должен выполнить следующее: найти лексему 'tata' и записать результаты в файл (поскольку другие аргументы не указаны, они сохраняют значения по умолчанию). Результат работы программы представлен в табл. 1.

Таблица 1: Пример выдачи функции `pol_search`

| | left | center | right |
|---|---|---|---|
| 0 | pieczołowicie, jak kiedyś mój | tata | . Noemi przyrządziła pyszny |
| 1 | ojcem ani wymówek, że | tata | jest zbyt surowy, a |
| 2 | że ten dzidziuś to wykapany | tata | ! - usłyszałam za |
| 3 | za sobą. - Czyj | tata | ? - nie zrozumiał mąż |
| 4 | m mężczyznę. Był to | tata | Rafałka. Siedział na urlopie |
| 5 | tracić takich pieniędzy. Nianiek | tata | Rafałka nie uznawał. Wiedział |
| 6 | .. - zdziwił się | tata | Rafałka. Dwie mamy spojrzały |
| 7 | dziadków na wieś, a | tata | Rafałka zabrał rodzinę do |
| 8 | . - Kto to jest | tata | Rafałka? - zainteresowała się |
| 9 | - No... | tata | Rafałka. - Aha - |
| 10 | Rafałka. - Kim jest | tata | Rafałka? - No. |
| 11 | Podupadłe gospodarstwo wiejskie znalazł nam | tata | Rafałka. Wiekowa chałupa w |
| 12 | po zakupy, ale gdzie | tata | ? Zawsze słucha porannego |
| 13 | tłoczyli się: mama i | tata | Kosmy Szczęściarza - tata w |
| 14 | i tata Kosmy Szczęściarza - | tata | w mamy piżamowej bluzce w |

# 3 Проблемы при создании API

Основные проблемы, с которыми пришлось столкнуться во время разработки, относятся к архитектуре различных корпусов. Во всех рассмотренных корпусах

нет возможности скачать выдачу в удобном для пользователя формате либо функции скачивания нет вообще. Например, в НКРЯ можно скачать выдачу только в XML (и только 1000 примеров, регулировать их количество нельзя), а в NKJP и DWDS такой возможности вообще нет. Эту проблему частично решает созданный инструмент.

Другая проблема — использование JavaScript в архитектуре корпусов. Наличие JS сильно усложняет парсинг. Например, в Чешском Национальном Корпусе (см. `https://www.korpus.cz`) и в Английско-Китайском параллельном корпусе (см. `http://www.jukuu.com`) используется JS для генерации KWIC-выдачи. Также усложняет работу отсутствие некоего "режима для разработчика"– облегченной версии сайта или даже возможности задавать запросы напрямую в базу и получать оттуда результаты, минуя промежуточные преобразования в html оболочку. Такая версия сайта среди рассматриваемых корпусов есть только в НКРЯ, что позволяет извлекать морфологические теги. При отсутствии такой версии у корпусов, которые используют JS, процесс парсинга сильно усложняется, поскольку на веб-странице с выдачей все теги находятся в JS-окне.

Морфологическая разметка также есть не везде: среди рассматриваемых корпусов такой уровень разметки есть только в NKJP и в НКРЯ: в DWDS, например, разметка отсутствует полностью. Еще одним немаловажным различием является отсутствие работы с регулярными выражениями. Например, в DWDS их нет совсем, а в НКРЯ и NKJP язык регулярных выражений несколько различается. К примеру, для того чтобы найти слово, начинающееся на букву *д*/*d*, в НКРЯ запрос должен выглядеть следующим образом: в НКРЯ — *д**, в NKJP — *d.**. Такие различия также мешают унификации запросов, т.к. пользователю все равно придется разбираться в системе регулярных выражений отдельно для каждого корпуса.

Стоит заметить, что из-за отсутствия унификации на уровне выдачи извлечение информации, отличной от выдачи (метатекстовая, перевод и тд.) сильно затруднено. Возможно, что при использовании унифицированного шаблона (см. van Gompel and Reynaert [2013]), извлечение такой информации не представляло бы никаких проблем.

# 4 Заключение

Созданный в результате работы пакет позволяет работать с корпусами НКРЯ, NKJP и DWDS, интегрировать результаты работы в различные исследовательские программы. Количество корпусов, доступных для обработки будет расширяться в дальнейшем. Возможность дальнейшей обработки результатов с использованием многофункционального пакета Pandas существенно расширяет рамки использования пакета, возможность сохранения результатов облегчает работу с данными. Кроме того, пакет хорошо вписывается в парадигму воспроизводимо-

сти научных исследований: если раньше процесс сбора данных из корпусов являлся отдельной задачей, то теперь при помощи пакета `lingcorpora` этот процесс можно интегрировать в получающие все большую популярность интерактивные документы, показывающие ход эксперимента. С интеграцией и автоматизацией сбора корпусного материала также легко связать инструменты обработки естественного языка, например, разного рода парсеры.

Также разработка проекта была бы существенно облегчена, если бы все корпуса поддерживали некоторый единый шаблон представления лингвистических примеров. Такой шаблон предлагается, например, в работе van Gompel and Reynaert [2013], и представляет собой многослойную XML-разметку. Некоторые лингвистические программы, такие как ELAN (`http://tla.mpi.nl/tools/tla-tools/elan/`), см. также Wittenburg et al. [2006]) и Fieldworks SIL (`http://fieldworks.sil.org`), также предоставляют возможность экспорта в формате XML-шаблонов, переводимых один в другой. Если бы тот или иной шаблон с множеством полей (скрытых от пользователя, в случае, если они не заполнены) использовался в большинстве корпусов, процесс сбора языковых данных был бы значительно облегчен, а получаемые результаты легко было бы сводить друг с другом и группировать в новую базу. Кроме того, такой шаблон позволил бы впоследствии переходить к созданию некоторой объединенной базы многоязычного корпуса.

## Литература


Chu-Ren Huang, Adam Kilgarriff, Yiching Wu, Chih-Ming Chiu, Simon Smith, Pavel Rychly, Ming-Hong Bai, and Keh-Jiann Chen. Chinese sketch engine and the extraction of grammatical collocations. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pages 48–55, 2005.

Wolfgang Klein and Alexander Geyken. Das digitale Wörterbuch der deutschen Sprache (DWDS). *Lexicographica*, 26:79–93, 2010.

Adam Przepiórkowski, Rafal L Górski, Barbara Lewandowska-Tomaszyk, and Marek Lazinski. Towards the National Corpus of Polish. In *LREC*, 2008.

Maarten van Gompel and Martin Reynaert. FoLiA: A practical XML Format for Linguistic Annotation-a descriptive and comparative study. *Computational Linguistics in the Netherlands Journal*, 3:63–81, 2013.

Peter Wittenburg, Hennie Brugman, Albert Russel, Alex Klassmann, and Han Sloetjes. Elan: a professional framework for multimodality research. In *Proceedings of LREC*, volume 2006, page 5th, 2006.

АМ Молдован. Национальный корпус русского языка. *Вестник Российской академии наук*, 77(6):498–504, 2007.

## Компьютерные ресурсы

Национальный Корпус Польского Языка: `http://nkjp.pl`

Национальный Корпус Русского Языка: `http://www.ruscorpora.ru`

Страница проекта на GitHub: `https://github.com/alexeykosh/lingcorpora.py`