10.15452/SlavicaIuvenum.XXI.02

Илья АФАНАСЬЕВ

КОРПУС СТАРОСЛАВЯНСКОГО ЯЗЫКА: НЕДОСТАЮЩЕЕ ЗВЕНО В ДИАХРОНИЧЕСКОЙ СЛАВИСТИКЕ

The Old Church Slavonic Corpus: a Missing Link in Historical Slavic Studies

***Keywords**: corpus linguistics, Slavic studies, Old Church Slavonic corpus, tokenization, historical linguistics, Old Church Slavonic*

***Contact**: СПбГУ; st079549@student.spbu.ru*

1 Предпосылки создания

1.1 Постановка проблемы

Исследование концептуальной оппозиции «свой – чужой» в старославянском языке активно продолжается на протяжении долгих лет (Вендина 2002: 38). Тем не менее, в ходе непосредственного изучения возникает проблема, связанная с источниками. А именно: основными являются словарь под ред. Й. Курца, начавший издаваться в середине XX в. (Kurz 1954), и словарь под ред. Р. М. Цейтлин, Э. Вечерки и Э. Благовой, вышедший в один год с последним томом словаря под ред. Й. Курца (Цейтлин и др. 1994). Изучение концептуальной оппозиции, объекта, очень тесно связанного с живой речью и пониманием мира, «Weltanschaung» (Kant 2008: 99), по описанию лексики языка – занятие затруднительное. Верно, что в обоих словарях авторы стараются дать достаточное количество контекстов, однако иногда тех не хватает для полноценного установления принадлежности слова к концептуальной оппозиции «свой – чужой». Обычно в такой ситуации во многих других языках возможно обратиться к корпусу для расширения нужного контекста или просмотра дополнительных. Однако исследователь старославянского языка ограничен и здесь. Ряд существующих корпусов не включают в себя полного объема текстов (PROIEL) (Общежитие). Большая часть текстов представлены как raw text (загруженные на сервер обычные .txt-файлы), что оставляет единственным инструментом поиска уже встроенный в браузер или текстовый редактор: зависит от программы, которой исследователь открывает файл (TITUS). В конце концов, в самом полном корпусе тексты закодированы в ASCII, а не Unicode: используется латинский

алфавит с более чем десятью техническими символами (ССМН). Узнать в этом старославянский текст очень тяжело; сформировать нужный запрос с учетом графических особенностей еще тяжелее. Такие данные возможно использовать в когнитивных исследованиях, однако с этим связано слишком большое число затруднений, и они все еще не дают достаточно полной картины. Необходимо спроектировать нечто хотя бы более удобное, а в отдаленной перспективе – более полное и предоставляющее еще больше возможностей для исследователя (вплоть до семантической разметки).

1.2 Предполагаемое решение проблемы

Это нечто – полнофункциональный корпус старославянского языка, который «…отличает (…) от электронной библиотеки – возможность настройки разнообразных параметров поиска, в том числе и синтактико-грамматических» (Архангельский, Кисилиер 2018: 51). Целью создать его мы и задались. Однако это очень масштабная цель, изложить результаты реализации которой в рамках статьи все же достаточно тяжело, и мы хотим сосредоточиться на конкретной части и лингвистических проблемах, с которыми мы столкнулись при работе над ней. Эта часть – токенизация предобработанного текста, полученного из существующих источников путем их парсинга (ССМН). Под токенизацией в данном случае мы подразумеваем в первую очередь «определение границ слов», хотя в дальнейшем, бесспорно, планируем провести и «демаркацию клитик, единиц из нескольких слов, аббревиатур и чисел» (Attia 2007: 65). Процесс будет состоять из написания соответствующего комплекса функций для уже существующей программы, занимающейся предобработкой данных, которые после будут переданы в модули разметки для непосредственной работы с ними. Результат работы – токенизированный текст. Примеры будут приведены из Пражских фрагментов, одного из текстов, который планируется включить в итоговый корпус (ССМН).

1.3 Структура

Первый раздел работы служит для постановки проблемы (п. 1.1), предложения способа ее решения (п. 1.2) и описания структуры (п. 1.3). Во втором разделе описываются выбранные для корпуса тексты и обосновывается необходимость выбора именно их (п. 2.1). Затем обсуждаются проблемы в области кодировки и графики, широко представленные в текстах (п. 2.2). Исследователи, занимавшиеся оцифровкой последних и помещением в Интернет, предложили

свои решения данных вопросов, которые мы считаем необходимым обсудить в рамках нашей работы (п. 2.3). Затем мы переходим к токенизации предобработанного текста и демонстрации ее результатов (п. 2.4). Наконец, в третьем разделе подводится итог нашей работы на конкретном этапе.

2 Корпус старославянского языка: в путь к разработке

2.1 Отбор текстов

В ходе отбора текстов исследователю приходится руководствоваться двумя наиболее важными критериями: релевантности относительно задачи, которую планируется решить созданием корпуса, и доступности самих текстов. Первый формирует оптимум (набор текстов, наиболее подходящий для решения задачи), второй – максимум (эти тексты, а также те, которыми можно дополнить соответствующий набор без потерь для эффективности исследования). В нашем случае оптимумом предполагаются тексты, использованные при создании «Старославянского словаря», комплекс наиболее древних памятников, таких как Зографское четвероевангелие, Мариинское четвероевангелие, Ассеманиево (или Ватиканское) евангелие-апракос, Киевские листки, Клоцов сборник, Супрасльская рукопись и др. (Цейтлин и др. 1994: 13). Максимумом – те из этих текстов, которые удастся обнаружить в открытом доступе, а также ряд значимых переходных между старославянским (здесь точнее будет использовать термин «староцерковнославянский», подразумевая временной промежуток с X по XI вв.) и церковнославянским (*sensu stricto*: более поздние тексты, испытывавшие влияние соответствующих славянских языков на ареалах своего распространения) языками текстов, таких как Пражские листки, «первая рукопись чешского извода церковнославянского языка» (ССМН). Решение о включении последних на данном этапе создания корпуса требует дальнейшего лингвистического обоснования, в том числе через уточнение понятий «старославянский язык», «церковнославянский язык» и «староцерковнославянский язык», а именно уточнение той сущности, которая за первым и последним из этих понятий скрывается (Kamphuis 2020: 1–17).

2.2 Вопросы предобработки

На данный момент не существует не просто цельного корпуса языка, но даже единой электронной коллекции текстов, на основании которой предстоит его

собрать. Когда же это произойдет, нам предстоит решить вопрос, связанный с непосредственным представлением данных: какую кодировку использовать?

Мы располагаем двумя основными вариантами: это ASCII и Unicode. К преимуществам первого относится поддержка на наибольшем возможном количестве устройств, к преимуществам второго – более адекватное отображение старославянского текста: кириллическое с поддержкой диакритических знаков над буквами. Недостатки каждого полярны преимуществам другого. Текст в кодировке ASCII выглядит совершенно не так, как следует выглядеть старославянскому тексту: «…zakonu ot& uCenika…» (ст.-слав. «…законоу отъ ученика…», рус. «закону от ученика») (CCMH). Unicode поддерживается не всеми машинами, соответствующие символы возможно найти не во всех шрифтах. Но в любом случае выбор кодировки подразумевает выбор графической системы: латиницы или кириллицы.

Этот выбор варьируется от исследователя к исследователю и от эпохи создания корпуса к эпохе.

2.3 Краткий обзор подходов к решению проблем предобработки

Большая часть материала (включая как достаточно крупные (Супрасльская рукопись и др.), так и значительно уступающие им по объему источники (Киевские листки и др.)) доступна в Интернете, правда, не всегда в своем изначальном виде, а либо уже в практически полностью обработанном Unicode-формате (PROIEL), либо в транслитерации на латинский алфавит для удобства представления в кодировке ASCII (CCMH). Это и есть два основных подхода к разрешению проблемы кодировки и графики. Консенсуса пока не достигнуто.

Однако, во-первых, более современный корпус (PROIEL) использует кириллицу на всем пути текста от дешифрованного черновика до представления другим исследователям. Во-вторых, даже изначально латинизированные тексты (CCMH) постепенно подготавливаются к кириллическому изданию (TITUS). В-третьих, за последние тридцать лет (столько прошло с формирования первых корпусов старославянского языка (CCMH)) кодировка Unicode стала поддерживаться на большем количестве персональных компьютеров, и принципиальной разницы между ее использованием и использованием ASCII нет, при этом понимание текста исследователям она облегчит. Мы не упоминаем уже о том, что исследователь, ссылающийся на корпуса в кодировке ASCII, все равно вынужден проводить транслитерацию вручную. Исходя из этих обстоятельств, мы склоняемся к использованию кириллической графики

и Unicode. Это подводит нас к двум путям взаимодействия с существующими электронными коллекциями старославянских текстов, варьирующимся в зависимости от решений, принимаемых разработчиками при создании.

В первом случае при предобработке будет достаточно загрузки данных, предоставляемых по лицензии, разрешающей свободное некоммерческое использование с указанием источника (PROIEL). К сожалению, так обработаны далеко не все тексты, и сама обработка для некоторых неполна (это и породило необходимость нашего текущего исследования), поэтому приходится обращаться к иным коллекциям.

Они представляют затруднения более значительные. Здесь исследователи, руководствуясь принципами доступности, дешифровали исходный текст в латиницу, к тому же, восстановив этимологические редуцированные, разбив тексты по особым образом нумерованным строкам и расставив знаки переноса. Эти решения и работа с результатами их принятия составили предмет отдельной части исследования, и к данному его моменту мы их затрагивать не будем, считая, что работаем с полностью предобработанным текстом, имеющим следующий вид (некоторые технические значки оставлены для более точного визуального представления в дальнейшем): «B' CBЪTiЛЪHЪ . (…) ==========ер =========овет========== !ги» (рус. «Поется в светилен. (далее – неразборчивые отрывки) … господи») (ССМН). Следующий шаг – разделение текста на токены, границы которых, как правило, совпадают с графическими границами слов, что облегчает работу исследователя по сравнению с рядом других языков: в урду, например, «буква изменяет свою форму в соответствии со следующей буквой» (Daud и др. 2017: 283).

2.4 Токенизация текста

В качестве иллюстрации к манипуляциям, проводимым на самом корпусе, мы используем строку из Пражских листков (ССМН): «B' CBЪTiЛЪHЪ . (…) ==========ер =========овет========== !ги» (рус. «Поется в светилен. (далее – неразборчивые отрывки) … господи») (ССМН).

Текст, хотя и обладал целым рядом графических особенностей (знак равенства используется для обозначения неразборчивости), по нашей догадке не должен был вызвать больших сложностей у токенизатора, справляющегося с русским языком. Мы решили инкорпорировать в нашу программу скрипт на языке программирования Python (Python Software Foundation), использующий возможности пакета NLTK (Bird и др. 2009), а точнее функцию

word_tokenize(string), где в переменную string была бы помещена наша строка. Результат – такой набор токенов: 'В', """, 'СВѢТiЛЪНѢ', '.', '==========ер', '=========овет==========', '!', 'ги'. Как можно заметить, в отдельные токены оказалась помещены все неалфавитные символы, в том числе технические значки и паерок, все квадратные скобки, поставленные исследователями, переводившими текст в цифровую форму. Этот инструмент нас не удовлетворил, поскольку разделял строки на слишком маленькие части.

Тогда мы обратились к основным средствам языка C#, на котором написана наша программа (Microsoft Corporation). Мы использовали метод Split() класса String, указав в качестве разделителя пробел и применив на нашей строке. Массив получился несколько иным: В', СВѢТiЛЪНѢ, . , ==========ер, =========овет==========, !ги. Квадратные скобки не были выделены в отдельные токены. Данный вариант, хотя был значительно проще, нас удовлетворял в большей степени, но все же не до конца. Мы хотели, чтобы маркеры фрагментарности не искажали компьютерное представление слов для последующей обработки. При этом оставить их было необходимо для сохранения понимания того, что текст собой представляет. Поэтому мы, внимательно его изучив, разработали следующий алгоритм:

А. Если строка начинается с маркера фрагментарности «=», то она разделяется на любые по длине последовательности любых символов и любые по длине последовательности маркеров фрагментарности «=». Каждая из этих последовательностей является токеном.

Б. Если строка начинается с любого иного символа и может быть приведена к общему виду «начало строки – любое число любых символов – любое число маркеров фрагментарности " = " – любое число любых символов – конец строки», то она является токеном.

В. Если строка начинается с любого иного символа, и в ней не присутствуют последовательности маркеров фрагментарности «=» длиной строго больше двух, то она является токеном.

Г. В любом другом случае строка разделяется на любые по длине последовательности любых символов, могущие включать маркер фрагментарности «=» в количестве двух единиц или меньше, и последовательности маркеров фрагментарности «=» длиной строго больше двух. Каждая из этих последовательностей является токеном.

Применив соответствующие паттерны, мы путем использования метода Split() класса Regex (Microsoft Corporation) получили следующие токены или их наборы для каждого из указанных выше случаев:

А. ==========; ер (ССМН)

Б. просв=ць (ССМН)

В. сво= (ССМН)

Г. свѣтъ==, облац, ====, ѧ(ѧ, ====(ССМН)

Как мы можем видеть, строка-пример выглядит следующим образом и, как и весь текст, готова к процедуре частеречной разметки:

В'

СВЂТiЛЪНЂ

==========

ер

=========

овет

============

!ги

Как можно видеть, алгоритм все еще не совершенен, и, возможно, на большом объеме текстов откроются дополнительные проблемы, но для анализа Пражских листков его достаточно.

Проблемы представляют такие токены, как «риноу!т!р!а!пезы» (ССМН) (в исходном тексте не оказалось нужного пробела), «м'но=====рѣха» (ССМН) (в силу плохой сохранности текста совершенно не очевидно, сколько здесь представлено слов) и «ъ==вотъ» (ССМН) (слишком малое количество маркеров фрагментарности «=»: при распространении правила (В) на случаи такого рода возникнет значительно больше ошибок; при этом очевидно, что мы имеем дело с двумя разными словами). Каждый из этих случаев ставит под вопрос применение автоматизации для разделения текста на токены, но, как нам кажется, каждый возможно разрешить применением словаря. В первом типе распознается «трапезы», в третьем – «вотъ», во втором программа придет к невозможности комбинации начала и конца и также разделит токен на набор.

Однако применить словарь в данный конкретный момент представляется невозможным: недостаточно уже обработанных текстов, на базе которых можно формировать соответствующую базу данных. Это станет одной из задач нашего исследования на следующем этапе.

3 Заключение

Таким образом, столкнувшись с нехваткой необходимого для когнитивных исследований материала, мы начали разработку корпуса старославянского языка. Для создания обработчика и токенизатора текста мы проанализировали опыт предыдущих исследователей, выделив ключевые проблемы, а именно графические различия представленных в Интернете текстов и фрагментарность некоторых из них. Помимо этого, определены границы токена и проведено разделение текста Пражских листков в соответствии с предложенными определением и методикой. Новым шагом для нас станут использование программы на иных текстах, а также постепенное проведение частеречной разметки уже обработанных.

Summary

In the article a way to deal with an absence of a proper Old Church Slavonic Corpus is proposed. Firstly, a problem is stated, and some comments on the previous research are given. It is followed by choice of the texts for the corpus description, preprocessing issues and the author’s approach to them declaration, after which a method of tokenizing the text is provided. The conclusion summarizes the results and states the next steps of the research.

Литература

Архангельский, Т. А., Кисилиер, М. Л. Корпуса греческого языка: достижения, цели и задачи. *Индоевропейское языкознание и классическая филология*. 2018 (22/1), с. 50–59.

Вендина, Т. И. *Средневековый человек в зеркале старославянского языка*. Москва: Индрик, 2002.

Общежитие. Режим доступа: http://www.obshtezhitie.net (2020-04-12).

Цейтлин, Р. М., Вечерка, Р., Благова, Э. *Старославянский словарь (по рукописям X–XI веков)*. Москва: Русский язык, 1994.

Attia, M. A. Arabic Tokenization System. In: *Proceedings of the 5th Workshop on Important Universal Matters*. Madison: Omnipress, 2007, s. 65–72.

Bird, S., Loper, E., Klein, E. *Natural Language Processing with Python*. Sebastopol: O'Reilly Media Inc., 2009.

CCMH (Corpus Cyrillo-Methodianum Helsingiense).
Режим доступа: http://www.helsinki.fi/slaavilaiset/ccmh/index.html (2020-04-12).

Daud, A., Khan, W., Che, D. Urdu language processing: a survey. *Artificial Intelligence Review*. 2017 (47), s. 279–311.

Kamphuis, J. *Verbal Aspect in Old Church Slavonic*. Leiden: Brill, 2020.

Kant, I. *Kritik der Urteilskraft*. Berlin: Akademie Verlag, 2008.

Kurz, J. *Slovník jazyka staroslověnského*. Praha: ČSAV, 1954.

Microsoft Corporation. *C# Language Specification Version 8.0*.
Режим доступа: https://docs.microsoft.com/en-us/dotnet/csharp/ (2020-04-14).

PROIEL. Режим доступа: http://foni.uio.no:3000/ (2020-04-12).

Python Software Foundation. *Python Language Reference, version 3.7*.
Режим доступа: http://www.python.org (2020-04-14).

TITUS (Thesaurus Indogermanischer Text- und Sprachmaterialien).
Режим доступа: http://titus.uni-frankfurt.de/indexe.htm (2020-04-12).