Санкт-Петербургский государственный университет

**АФАНАСЬЕВ Илья Андреевич**

**Выпускная квалификационная работа**

**Лингвистические проблемы корпуса старославянского языка**

Уровень образования: магистратура

Направление 45.04.01 «Филология»

Основная образовательная программа ВМ.5831 «Славянские языки и литературы»

Научный руководитель:

доцент, Кафедра славянской филологии СПбГУ,

Бабанов Андрей Владимирович

Научный консультант:

профессор, Факультет филологии НИУ ВШЭ,

Ляшевская Ольга Николаевна

Рецензент:

профессор, главный научный сотрудник, ФГБУН Института лингвистических исследований РАН,

Соболев Андрей Николаевич

Санкт-Петербург

2021

**Содержание**

[**Введение** 4](#_Toc72902568)

[**Глава 1. Современный опыт разработки текстовых корпусов и средств автоматической обработки для них.** 9](#_Toc72902569)

[**1.1.** Общая характеристика развития корпусной лингвистики в конце XX – начале XXI вв. 9](#_Toc72902570)

[**1.1.1.** Старославянский язык. 9](#_Toc72902571)

[**1.1.2.** Индоевропейские языки. 12](#_Toc72902572)

[**1.1.3.** Языки других семей. 15](#_Toc72902573)

[**1.1.4.** Современная корпусная лингвистика и исследования старославянского языка: перспективы соразвития. 19](#_Toc72902574)

[**1.2.** Подготовительный этап. 20](#_Toc72902575)

[**1.2.1.** Сбор текстов. 20](#_Toc72902576)

[**1.2.2.** Графическое представление. 22](#_Toc72902577)

[**1.2.3.** Токенизация. 24](#_Toc72902578)

[**1.3.** Частеречная разметка. 28](#_Toc72902579)

[**1.3.1.** Общая характеристика частеречной разметки как лингвистической проблемы. 28](#_Toc72902580)

[**1.3.2.** Набор тэгов: самостоятельная разработка или заимствование? 30](#_Toc72902581)

[**1.3.3.** Тэггеры: характеристика текущего состояния. 32](#_Toc72902582)

[**1.3.4.** Частеречная разметка как сфера поисков исследователя. 37](#_Toc72902583)

[**1.4.** Лемматизация. 37](#_Toc72902584)

[**1.4.1.** Определение лемматизации и основные области её применения в корпусной лингвистике. 37](#_Toc72902585)

[**1.4.2.** Лемматизация как задача автоматической обработки естественного языка. 38](#_Toc72902586)

[**1.4.3.** Лемматизаторы старославянского языка. 40](#_Toc72902587)

[**1.4.4.** Лемматизация при создании корпуса старославянского языка. 40](#_Toc72902588)

[**1.5.** Выводы. 41](#_Toc72902589)

[**Глава 2. Базовая теоретическая характеристика языка текстов старославянского канона.** 43](#_Toc72902590)

[**2.1.** Старославянский язык: определение, особенности, периодизация. 43](#_Toc72902591)

[**2.2.** Графические системы, репрезентирующие тексты канона старославянского языка. 48](#_Toc72902592)

[**2.3.** Частеречный состав старославянского языка. 52](#_Toc72902593)

[**2.4.** Выводы. 55](#_Toc72902594)

[**Глава 3. Старославянский язык как объект составления корпуса.** 57](#_Toc72902595)

[**3.1.** Определение текстового состава корпуса старославянского языка. 57](#_Toc72902596)

[**3.2.** Автоматическая предобработка некоторых текстов старославянского языка. 76](#_Toc72902597)

[**3.3.** Частеречная разметка документов в корпусе старославянского языка. 79](#_Toc72902598)

[**3.4.** Лемматизация токенов старославянского языка. 90](#_Toc72902599)

[**3.5.** Выводы. 95](#_Toc72902600)

[**Глава 4. Программная реализация корпуса старославянского языка.** 97](#_Toc72902601)

[**4.1.** Модуль предобработки. 97](#_Toc72902602)

[**4.2.** Модуль представления. 107](#_Toc72902603)

[**4.3.** Выводы. 111](#_Toc72902604)

[**Заключение** 112](#_Toc72902605)

[**Список использованной литературы** 116](#_Toc72902606)

## **Введение**

За то время, пока данная работа находилась в процессе написания, из сети исчезло несколько корпусов старославянского языка. Последним из них ненадолго стал Corpus Cyrillo-Methodianum Helsingiense, корпус старославянского языка университета Хельсинки [CCMH]. Этот корпус и работа с ним были главным источником вдохновения при написании данной работы. Сама возможность его исчезновения сделала эту работу существенно более важной, чем казалось в момент её начала: в данный момент проблемой выступает не только создание нового корпуса, но и сохранение уже существующих.

Корпусы языков с небольшим количеством известного текстологического материала, к каковым относится старославянский, уязвимы: понимание этого сейчас сильно, как никогда, как и понимание того, что некоторые корпусы, возможно, придётся пересоздавать.

**Актуальность** данной работы обусловлена необходимостью репродукции и усовершенствования существующих электронных корпусов старославянского языка. **Теоретическое значение** работы заключается в описании процесса создания корпуса старославянского языка и проблем, которые возникают в ходе данного процесса в связи со специфическими чертами старославянского языка как языкового идиома (как следствие, теоретическое значение работы состоит в том числе в лингвистическом описании данных черт).

Исследовательская гипотеза может быть сформулирована следующим образом: старославянский язык, обладая достаточно высокой гетерогенностью лингвистических особенностей конкретных своих реализаций, может быть обработан и помещён в корпус как единый идиом, представленный некоторым количеством наиболее близких лингвистически текстов.

На защиту выносятся следующие положения:

1. Существующие ресурсы для исследования старославянского языка нуждаются в развитии и пополнении.
2. Тексты для корпуса старославянских языков необходимо отбирать по строго определённым, формализованным лингвистическим критериям. В работе используются критерии, данные в различных теоретических работах по старославянскому языку, в частности, исследовании [Kamphuis, 2020].
3. Не все тексты, прежде определённые как старославянские, соответствуют данным критериям.
4. Старославянский канон отличается высокой степенью лингвистической гетерогенности [Поливанова, 2013, XV], что влияет на эффективность применения методов машинного обучения при его разметке.
5. При адаптации существующих методов машинного обучения возможно создать модели, способные, обучившись на одном тексте старославянского языка, успешно размечать остальные.

Прежде многие работы характеризовали существующие корпусы старославянского языка, однако непосредственно процесс создания, в совокупности с описанием индивидуального подхода к решению задач, возникающих в ходе него, достаточно подробно рассматривается впервые. Это обуславливает **новизну** работы.

**Практическая значимость** работы состоит, в первую очередь, в создании корпуса старославянского языка на базе универсальной системы создания корпусов. Этот корпус будет возможно использовать в дальнейших лингвистических исследованиях [Egbert и др., 2020], моделировании языка на основании узуса [Divjak и др., 2017, с. 177], лингводидактике [Romer, 2011], а также при составлении словарей [БФССЯ, 2021].

Целью является выявление и решение лингвистических проблем, с которыми исследователь сталкивается при создании корпуса старославянского языка. К числу таких проблем могут быть отнесены, в частности, определение текстового состава и выбор моделей машинного обучения для осуществления автоматической разметки.

Задачами работы, соответственно, становятся:

1. Анализ опыта создания электронных корпусов старославянского языка, а также актуальных подходов к созданию корпусов и обработке естественного языка (на данном этапе конкретных её областей: предобработки, токенизации, частеречной разметки и лемматизации) в современной лингвистике.
2. Характеристика старославянского языка как языкового идиома, а также его особенностей, которые необходимо учитывать при создании корпуса.
3. Структуризация информации о проблемах, которые лингвист-исследователь должен решить при создании корпуса старославянского языка, с учётом особенностей последнего как языкового идиома.
4. Описание способов решения лингвистических проблем, возникающих при создании корпуса старославянского языка.
5. Описание процесса создания корпуса старославянского языка.

Работа состоит из 4 глав.

В первой главе характеризуется состояние корпусной лингвистики в начале XXI века (параграф 1.1; подпараграфы описывают корпусы конкретных языков, а именно старославянского (1.1.1), других индоевропейских языков (1.1.2), языков других семей (1.1.3); также подводятся краткие итоги (1.1.4)). В параграфе 1.2. даётся описание существующих методик предварительной обработки текстов (сбор текстов в подпараграфе 1.2.1, графическое представление в подпараграфе 1.2.2, токенизация в подпараграфе 1.2.3). Параграф 1.3 предлагает характеристику подходов к частеречной разметке (общую в подпараграфе 1.3.1, особенности формирования набора тэгов в 1.3.2, обзор тэггеров в 1.3.3, краткие выводы в 1.3.4). В параграфе 1.4 характеризуются подходы к лемматизации (лемматизация как лингвистеская задача описана в подпараграфе 1.4.1, как задача обработки естественного языка – в подпараграфе 1.4.2, существующие лемматизаторы старославянского языка – в подпараграфе 1.4.3, краткие выводы даны в подпараграфе 1.4.4). Параграф 1.5 представляет собой промежуточные выводы по данному разделу.

В главе 2 описывается старославянский язык как языковой идиом (параграф 2.1), репрезентирующие его графические системы (параграф 2.2), его частеречный состав (параграф 2.3), а также даётся общее описание сложностей, возникающих при работе с его текстами (параграф 2.4).

Глава 3 концентрируется на лингвистическом анализе решения проблем, с которыми исследователь сталкивается при создании корпуса старославянского языка, а именно – выборе текстов (параграф 3.1), графической предобработке (параграф 3.2), частеречной разметке (параграф 3.3) и лемматизации (параграф 3.4). В конце третьей главы указывается на положительные и отрицательные стороны предложенных решений (параграф 3.5).

Глава 4 характеризует программную реализацию корпуса старославянского языка, а именно – модуль предобработки (параграф 4.1) и модуль представления (параграф 4.2).

Теоретическая база работы складывается из существующих исследований старославянского языка. Большая часть этих работ носит фундаментальный характер и была написана в середине двадцатого века: некоторые положения, в частности, отнесение тех или иных рукописей к старославянским, с тех пор были пересмотрены. На это указывается в современных работах, посвящённых старославянскому языку [Kamphuis, 2020].

Методологическую базу работы составляют исследования в области компьютерной лингвистики, которые разделены на две подгруппы. Первая посвящена созданию первых корпусов для различных языков мира. На основании опыта исследователей вычисляется оптимальный подход к созданию корпуса старославянского языка. Вторая подгруппа методологических исследований касается вопросов обработки естественного языка, а именно препроцессинга, токенизации, частеречной разметки и лемматизации. Путём их анализа выявляется, какие методы лучше всего применить для решения этих задач на материале старославянского языка.

В исследовании используются метод индукции (при определении текстового состава корпуса старославянского языка), метод анализа (при анализе текстов старославянского канона), метод эксперимента (при выборе инструментов обработки естественного языка) и описательный метод (при характеристике процедур, применяемых в ходе создания корпуса старославянского языка).

Объектом исследования выступает старославянский язык как языковой идиом, на материале которого потенциально создаётся корпус. Предметом исследования являются лингвистические особенности старославянского языка, которые необходимо учитывать при создании корпуса. Материалом исследования становятся все тексты, которые в той или иной момент времени считались текстами старославянского канона. Список текстов и их анализ будет приведён непосредственно в ходе работы.

## **Глава 1. Современный опыт разработки текстовых корпусов и средств автоматической обработки для них.**

Создание корпуса – многомерная задача, состоящая из целого ряда этапов, выполняющихся, как правило, последовательно. Каждый из этих этапов (предобработка, токенизация, частеречная разметка, лемматизация) становится существенным шагом для работы, а его результат становится фундаментом для дальнейшей исследовательской деятельности. Представить корпус, в котором хотя бы один этап был оставлен без внимания, затруднительно. И даже если это возможно, то далеко не всегда результат оказывается релевантным для лингвистики.

* 1. Общая характеристика развития корпусной лингвистики в конце XX – начале XXI вв.

Однако прежде чем обобщить накопившиеся знания в области корпусной лингвистики по вопросу непосредственно создания корпусов, необходимо уточнить общие тенденции её развития в последние двадцать лет. Иначе говоря: где сейчас создаётся больше всего текстовых корпусов, и исследователи каких языков высказывают наибольшую в них нужду?

**1.1.1.** Старославянский язык.

Начать следует с того языка, которым мы непосредственно занимаемся. В данном случае мы имеем дело с целым рядом ресурсов, в той или иной степени пытающихся представить этот язык в наиболее полном и репрезентативном виде, однако ни один из них не соответствует этим критериям в точности.

Наиболее полное собрание текстов представлено в Кирилло-Мефодиевском корпусе университета Хельсинки: Ассеманиево Евангелие, Мариинское Евангелие, Супрасльская рукопись, Зографское Евангелие, Саввина книга, а также представленные в более поздних списках жития Константина и Мефодия [CCMH]. Корпус является частью коллекции старославянских текстов Университета Гёте во Франкфурте, которая также включает в себя Киевские листки и Пражские фрагменты, памятник, традиционно считающийся старейшим текстом чешского извода церковнославянского языка и находящийся в пограничном положении между двумя стадиями развития одной и той же языковой системы. По всей видимости, основанием для его включения в коллекцию старославянских текстов было то, что он содержал в себе большее количество черт языка-предка, чем языка-потомка, был тем, что в биологии чаще всего называют переходной формой [Freeman, Herron, с. 84]. Точной причины составители коллекции Франкфуртского университета не называют, и лингвистическое сопоставление Пражских фрагментов и остального массива старославянского языка остаётся предметом будущего исследования.

И в Кирилло-Мефодиевском корпусе, и в коллекции старославянских текстов Университета Гёте не представлен целый ряд памятников совершенно разного объёма, в данный момент причисляемых к старославянскому канону, таких как Синайская псалтырь, и активно используемых в современных исследованиях [Kamphuis, 2020, с. 10]. Возникает необходимость дополнения корпуса существующими изданиями памятников, на данный момент не представленными ни в Кирилло-Мефодиевском корпусе, ни в коллекции старославянских текстов университета Гёте.

Схожими недостатками характеризуется и веб-приложение, разработанное в университете Осло и представляющее собой комплекс корпусов текстов Нового Завета на разных языках мира, в том числе и на старославянском [TOROT]. Его коллекция значительно расширилась за последние годы, она включает в себя Мариинское и Зографское Евангелие, Супрасльскую рукопись, Синайскую псалтырь, жития Кирилла и Мефодия, Синайский евхологий, а также Киевские листки.

Эти тексты подвергнуты частичной морфологической разметке. Выполнена эта разметка вручную, что делает её более точной, чем разметка многих других корпусов, однако на объёмах, значительно превышающих корпус университета Осло, её применение требует работы слишком большого количества специалистов в течение слишком большого количества времени. Однако для решения задачи, требующей построения достаточно полного и репрезентативного корпуса, и он оказывается подходящим не в полной степени.

В рамках проекта «Манускрипт» в данный момент собрано порядка десятка рукописей, входящих, по мнению ряда исследователей, в старославянский канон. Проведена их частеречная разметка и лемматизация, на их материале уже проводятся статистические исследования в области лингвистики [Баранов, 2019]. Однако тексты этой коллекции находятся в закрытом доступе (предоставляется только по запросу), и зачастую фрагментарны [Баранов, 2019, с. 4]. К тому же, большое количество текстов старославянского канона в «Манускрипт» на данный момент не включены. Сам проект нацелен на изучение древней славянской письменности в целом и, как следствие, не преследует целью создания инструментария для работы с непосредственно старославянскими текстами. А значит, не может считаться корпусом старославянского языка в полном смысле этого слова.

Все остальные текстовые корпусы старославянского языка, созданные за последние несколько десятков лет, на данный момент находятся в закрытом доступе, либо же вовсе исчезли из Интернета. Вопрос об их применении исчезает сам собой.

Существует также исторический корпус болгарского языка [Cyrillomethodiana]. Его тексты в наибольшей степени типологически похожи на тексты старославянского канона. Однако ни один из представленных в корпусе текстов традиционно как старославянский не определяется, поэтому и назвать сам корпус даже частично старославянским на данный момент невозможно.

Вместе с тем, в последние годы уже был проведён ряд исследований старославянского языка на базе корпуса, который исследователи оказались вынуждены собирать вручную. Примером такой работы может служить изучение глагольного вида в старославянском языке на базе корпуса [Kamphuis, 2020].

Таким образом, в исследованиях старославянского языка на данный момент наблюдается картина растущей необходимости в корпусном исследовании, совмещённая с отсутствием корпусов, способных эту необходимость удовлетворить. Для славистики это направление развивающееся, в корпусной лингвистике ему с каждым годом всё больше требуется ревитализация.

**1.1.2.** Индоевропейские языки.

Значительная часть крупных индоевропейских языков на данный момент уже снабжена текстовыми корпусами достаточной полноты и репрезентативности (ср., к примеру [Elenius и др., 2008] для шведского). Тем не менее, меньшие по количеству носителей или географии распространения языки представлены не так полно, как то требуется лингвистам, их изучающим. Это приводит к появлению новых комплексов корпусов [Streiter и др., 2004].

Так, для ирландского в последние десять лет создавался разговорный корпус, основным требованием которого было отражение «диалектной и диахронической вариативности» [Dhonnchadha и др., 2012, с. 1]. Корпус состоит из «записей радиоэфиров» и «частных разговоров четырёх «волонтёров», получив «итоговый объём в 140 000 слов» [Dhonnchadha и др., 2012, с. 2]. Корпус был транскрибирован, размечен по частям речи и опубликован на платформе SketchEngine [Dhonnchadha и др., 2012, с. 2 – 6], таким образом удовлетворяя существующий запрос.

Для болгарского языка, одного из ближайших современных родственников старославянского, на рубеже 2000-х и 2010-х был разработан семантически аннотированный корпус, предназначенный для обучения моделей, нацеленных на то, чтобы выполнять задачу разрешения лексической многозначности [Koeva и др., 2011]. Корпус состоит из «811 текстов, каждый длиной по 100+ слов, что в сумме составляет 101062 токена» [Koeva и др., 2011, с. 144]. В дальнейшем данный корпус будет расширяться, поскольку к данному моменту точность моделей на его базе составляет ~65% [Koeva и др., 2011, с. 149].

Постепенное исчезновение старой русской диалектной системы создаёт запрос на изучение того фонда, который удалось сохранить, необходимого для получения дополнительных знаний по лексике и истории русского языка. Тем не менее создание достаточно крупного корпуса пока ещё невозможно: от некоторых диалектов практически не осталось никакой информации, а количество всех диалектов слишком велико даже для крупной исследовательской группы; требуется внимание учёного объединения уровня института. Поскольку таковой на данный момент не создан, задача решается в различных научно-исследовательских учреждениях.

Так, в Саратове из записей диалогов с диалектоносителями создаётся Саратовский диалектный корпус диалектов д. Мегра и с. Белогорное [Крючкова, Гольдин, 2010]. Единица корпуса – ««запись» – расшифровка магнитофонной фиксации непрерывного фрагмента общения, приводимая во вспомогательной транскрипции – символьной записи, близкой к орфографической» [Крючкова, Гольдин, 2010, с. 5]. По проекту, «запись» размечается жанрово-тематически и морфологически (последнее – при помощи средств автоматической обработки текста); создаётся метаразметка каждой записи [Крючкова, Гольдин, 2010, с. 7]. Корпус на данный момент находится в разработке.

Разговорные и диалектные корпусы с их специфическим образом хранения манифестаций языковой системы в речи являются частью такого направления корпусной лингвистики, как разработка корпусов, дающих более полную картину языка как системы. Это корпусы, всё дальше отводящие корпусную лингвистику от принадлежности к исключительно текстологическим дисциплинам.

Примером корпусов такого рода может служить мультимедийный русский корпус, являющийся частью системы национального корпуса русского языка, к данному моменту уже достаточно активно разрабатываемый и хорошо описанный [Гришина, 2009]. Единицами такого корпуса всё ещё являются тексты, однако тексты, не существующие вне непосредственного контекста своего оригинального воспроизведения, и образующие с ним мультимедийное единство. Строго говоря, это всё та же «запись» диалектного корпуса, но расширенная до более крупных масштабов, к примеру, целого фильма. Мультимедийные корпусы при этом снабжены значительно более обширной системой метаразметки [Гришина, 2009, с. 189].

Некоторые корпусы полностью отходят от использования единиц языка, характеризующихся когезией и когерентностью на всём своём протяжении. Они изучают живую речь индивида в течение определённого количества времени: зачастую несвязную, постоянно прерывающуюся, иногда переходящую от одной коммуникативной интенции к другой. Это в большой степени отдаляет такие корпусы от традиционной парадигмы корпусной лингвистики, приближая их к максимально возможно полному изучению речевой деятельности человека. К таким корпусам относится, к примеру, проектируемый в Санкт-Петербурге корпус, позволяющий провести изучение «речевого поведения носителя языка в течение дня» [ОРД].

Таким образом, современная индоевропейская корпусная лингвистика, накопив достаточный опыт создания традиционных текстовых корпусов литературных языков, сосредоточилась на новом комплексе задач, диктуемых необходимостью исследовать, во-первых, не только тексты, но и спонтанную речь, во-вторых, взаимодействие языка с другими способами человеческой коммуникации и миром вокруг во время речевой ситации, в-третьих, диахроническую (предыдущие стадии развития) и синхроническую (территориальные диалекты, социолекты, варианты) вариативность языка. Для этого исследовательские группы прибегают к созданию диалектных и мультимедийных корпусов.

Для нашей работы последнее существенно в единственном смысле: корпусные исследования старославянского языка в значительной степени отстали от исследований других индоевропейских языков. Достижения современной индоевропеистики для старославянского языка на данный момент неприменимы (и, учитывая его специфику, возможно, не будут применимы никогда). Те методы, которые мы будем применять в своём исследовании, следует искать в работах, посвящённых другим языковым семьям и другим территориям.

**1.1.3.** Языки других семей.

Исследовательское сообщество арабского языка очень обширно и к текущему моменту создало достаточно большое количество корпусов, в том числе аннотированных [Khalifa и др., 2018], выработав при этом сложные модели работы с различными подсистемами языка. Всё это произошло за промежуток меньший, чем двадцать лет. Крупные корпусы арабского языка, собираемые исследователями с учётом конкретных задач, едва ли не моложе, например, последней даты обновления самых новых старославянских коллекций.

В течение последних двадцати лет было опубликовано значительное количество работ, указывавших на отсутствие корпусов арабского языка, а также на то, что качество исследований от недостатка корпусов «значительно страдает», даже в том случае, если каждый учёный «собирает свой собственный» [Mansour, 2013, с. 81]. Это замечание в целом универсально, и отнести его можно в том числе к старославянскому языку, который с арабским роднит гораздо больше, чем может показаться на первый взгляд, например, значительное количество диакритических знаков и «необычный алфавит», которые способствуют тому, что «в широких университетских кругах языком просто не занимаются» [Atwell и др., 2004]. В арабском, впрочем, иная ключевая проблема: исследователи концентрируются на частных корпусах вследствие того, что массив текстов, продуцируемый носителями арабского языка, очень уж велик и разнороден. Отмечается, что «существующие корпусы арабского языка слишком ограничены в выборе источников» [Atwell, Al-Sulaiti, 2006, с. 147].

Попытки, тем не менее, предпринимаются. Первые обобщающие труды по корпусам арабского языка появились уже сравнительно давно [Atwell, Al-Sulaiti, 2006]. Там же исследователи обозначали тип текста, необходимый корпусу современного арабского языка: «современный, доступный, арабский» [Atwell, Al-Sulaiti, 2006, с. 147]. Анонсирован корпус размером по меньшей мере в «один миллион слов» [Atwell, Al-Sulaiti, 2006, с. 166]. Авторы попытались добиться максимальной полноты и репрезентативности, однако этого к моменту написания ими работ достигнуто не было, в связи со «сложностью поиска ресурсов для всех возможных категорий текстов в вебе» [Atwell, Al-Sulaiti, 2006, с. 165].

Особое внимание уделяется учебным корпусам, которые, впрочем, по словам исследователей, могут использоваться для целого ряда других целей, в частности, тренировки нейросетей, занимающихся оптическим распознаванием знаков [Alfaifi, Atwell, 2013 A, с. 2]. На данный момент создан Учебный корпус арабского языка, состоящий из шести текстов и 1488 токенов, включающий в себя статистику самых частотных ошибок [Alfaifi, Atwell, 2013 B, с. 12].

Значительно больший блок составляют диахронические корпусы арабского языка. Запрос на них возник в последние десять лет, когда исследователями была озвучена необходимость появления «очень крупного корпуса классического арабского, через который следует понимать основы арабистики, который следует сделать доступным для лингвистов, которые могли бы сравнивать его с корпусами современного арабского (КСА), чтобы наблюдать за изменениями в языке, их обстоятельствами и причинами» [Alrabia и др., 2014, с .28].

Одним из корпусов, созданных исследователями, непосредственно заявлявшими о их необходимости, был «корпус классического арабского университета короля Сауда (ККАУКС, англ. KSUCCA)», содержащий «тексты от доисламской эпохи (…) вплоть до начала одиннадцатого века» [Alrabia и др., 2014, с .29]. ККАУКС состоит из 50602412 токенов, содержащихся в 410 документах шести жанров, а именно религиозного, лингвистического, литературного, научного, описания общественного устройства и биографического [Alrabia и др., 2014, с .29]. Корпус, таким образом, характеризуется значительным объёмом для исторического корпуса и сравнительно высоким жанровым разнообразием. Было продемонстрировано, что на материале корпуса можно было проводить исследования в изменении значений одного [Alrabia и др., 2014, с .34] или нескольких слов [Alrabia и др., 2014, с .30], устойчивых выражений [Alrabia и др., 2014, с .31] и словарного состава [Alrabia и др., 2014, с .33]. Помимо этого, корпус морфологически размечен при помощи средств автоматической обработки текста [Alrabia и др., 2014].

На территории одного из самых разнообразных по количеству языков континента, Африки, корпусная лингвистика также начала развиваться только на протяжении последних двадцати лет, однако исследователи уже достигли определённых успехов.

В обобщающих работах сравниваются корпусы по меньшей мере «25 субсахарских языков», некоторые из которых «содержат данные по 19 языкам» на момент написания статьи [Chiarcos и др., 2009, с. 17]. Тексты в этих корпусах размечены через систему ANNIS, которая позволяет осуществлять базовую морфологическую и синтаксическую разметку, механизм осуществления которой, впрочем, не совсем ясен из соответствующих работ [Chiarcos и др., 2009, с. 21]. Ключевой задачей, о которой упоминали исследователи, было «извлечение структуры информации, а именно структурирование лингвистической информации, чтобы оптимизировать перенос информации внутри дискурса: информацию необходимо подготовить («упаковать») различными путями в зависимости от целей, которые говорящий преследует в рамках этого дискурса» [Chiascos и др., 2009, с. 17]. Именно решение этой дискурсивной задачи в конечном итоге и привело к разработке ANNIS и интеграции через неё корпусов, содержащих данные по двадцати пяти языкам.

Отдельно на субсахарском языковом пространстве выделяется регион Южной Африки, где существуют несколько языков, число носителей которых и политическая важность в рамках региона приблизительно идентичны, что создаёт необходимость в «открытом ресурсе для педагогического применения и исследований» [Carstens and Eiselen, 2019, с. 65]. Для этого потребовалось «собрать данные из различных университетов, чтобы построить мультимодальный, мультижанровый, многоуровневый корпус текстов, созданных обучающимися, для которых языки корпуса являются или не являются родными, в университетах, на основе принятых в международном сообществе принципов построения корпусов» [Carstens and Eiselen, 2019, с. 68]. На данный момент корпус находится в начальной стадии своего построения. Цели, декларируемые авторами, создающими корпус, на данный момент не реализованы. Корпусная лингвистика на южноафриканском пространстве находится в начале развития, и уже к этому моменту она достаточно сильно специализирована, сосредоточена на педагогическом применении и аспекте создания корпусов для обучения языку, и в гораздо меньшей степени – его изучения.

Наконец, начинает развиваться корпусная лингвистика в изучении языков иных областей мира, к примеру, Индии [Singh, Bandyopadhyay, 2010] [Dash, Chaudhuri, 2001] или Южной Америки. Один из самых ярких показателей – создание корпуса языка паэс на территории Колумбии.

Цель, с которой создаётся данный корпус – ревитализация языка паэс через уже существующие тексты. Исследователи при работе задаются такими вопросами как «возможно ли получить письменные документы на паэс, хоть где-то? можно ли те, что доступны, использовать как что-то большее, чем просто электронный документ, могут ли они быть использованы для деятельности по ревитализации? как хорошо язык известен в письменной форме? Возможно ли создать инструменты, позволяющие развивать более сложные виды деятельности в ходе обучения паэс?» [Martinez и др., 2018, с. 882].

Выстраивание механизма ревитализации языка через корпус – задача весьма нетривиальная. Она подразумевает под собой сбор текстов, создание средств предобработки и разметки для унификации и придания минимальной полезности при обучении. Помимо этого, корпус должен быть максимально крупным, сбалансированным и репрезентативным: ровно настолько, насколько это возможно для корпуса вымирающего языка. Разработчикам корпуса паэс удалось использовать 8 текстов с 1176 словами и разметить их [Martinez и др., 2018, с. 888 – 889]. Эти числа достаточно маленькие по меркам языков более крупных, однако, по всей видимости, состояние языка накладывает свои ограничения. В полной мере корпус задачу ревитализации на данный момент решить не помогает. Тем не менее, исследователи настаивают на его будущем расширении в дальнейшей работе, равно как и на усовершенствовании набора тэгов и самого принципа разметки [Martinez и др., 2018, с. 892].

**1.1.4.** Современная корпусная лингвистика и исследования старославянского языка: перспективы соразвития.

Таким образом, состояние корпусной лингвистики старославянского языка на данный момент, несмотря на продолжительную историю, находится в ранней стадии развития.

В то время как индоевропейские языки сосредотачиваются на создании специализированных корпусов для достижения конкретных целей, задачу создания первого корпуса старославянского языка только предстоит решить. Задача параллельного представления как можно большего количества текстов, актуальная для субсахарского языкового пространства, частично решена проектом TOROT, однако при этом представленные в нём тексты не дают полной картины старославянского языка.

Наибольший интерес представляют в данный момент лингвистика арабского языка и лингвистика паэс. В первом случае ведётся активная разработка исторических корпусов на достаточно внушительном объёме текстов с использованием современных платформ. Во втором – создание первого корпуса для почти вымершего языка, разработка собственного набора тэгов и применение нестандартных методов обработки текста там, где обычные применимы в меньшей степени. При дальнейшем теоретическом обобщении накопленного опыта наибольший смысл имеет обращение к опыту исследователей именно этих языков.

* 1. Подготовительный этап.

**1.2.1.** Сбор текстов.

Различные исследователи при создании корпусов руководствуются различными критериями отбора текстов из различных источников.

Составители корпусов арабского языка, как правило, не рассматривают текстологический вопрос отдельно. В некоторых случаях делаются ссылки на опыт предыдущих работ и текстовые коллекции, которые собирались до корпуса, создание которого непосредственно описано в соответствующей работе [Alrabia и др., 2014, с. 30]. В других случаях, когда корпус становится первым в ряду, на источниках материала и вовсе стараются не акцентировать внимание, больше склоняясь к его квантитативной характеристике [Alfaifi, Atwell B, 2013, 11]. Подобный подход в целом тяжело критиковать с точки зрения полноты конкретного исследования: его проблема скорее в том, что методология науки в конечном итоге не пополняется, не растёт понимание того, по каким именно критериям происходит подбор материала., что в конечном итоге не даёт использовать его при обобщении и создании новых корпусов.

В случае с существующими корпусами старославянского языка, их оцифровка, равно как и размещение в электронных коллекциях, проводились непосредственно составителями корпуса по различным изданиям памятников с автоматически нераспознанным текстом [CCMH]. Этот подход гарантирует наибольшую корректность, однако при наличии уже существующих обработанных текстов значительно увеличивает необходимое на составление корпуса время, что в условиях, когда требуется оперативное составление коллекции и создание продукта минимальной полезности, не совсем релевантно.

Наконец опыт создания корпуса языка паэс показывает возможность обращения при проектировании к уже существующим текстовым коллекциям языка [Martinez и др., 2018, с. 888]. Такой подход полностью оправдывает себя в условиях, когда корпус необходимо собрать как можно быстрее, хотя и не избавлен от недостатков. Например, проблемы возникнут при графической унификации (для языка паэс, алфавит для которого был приведён к единому образцу только в 2000 году [Martinez и др., 2018, с. 882]), не говоря уже о том, что в некоторых случаях требуется сверка с оригиналом оцифрованного текста, не всегда доступным исследователю, составляющему корпус на основе предыдущих коллекций. Наконец, если текст подвергался какой бы то ни было трансформации, существует возможность возникновения разногласий между различными исследователями относительно механизма проведения этой трансформации: примером может служить вопрос об отражении фонетических различий в диалектных корпусах [Крючкова, Гольдин, 2010, с. 8 – 11]. В таких случаях обычно приходится обращаться к документации исходной коллекции, и если обнаруживается принципиальное несовпадение, устранять различия либо вручную, либо при помощи специально написанной программы. Тем не менее при малом количестве материала эта проблема не является существенной.

При составлении корпуса старославянского языка, таким образом, в наибольшей степени оказалось возможным ориентироваться на опыт исследователей языка паэс. Оптимальной тактикой будет составление корпуса из существующих коллекций (при условии распространения их по соответствующей лицензии) с возможной самостоятельной оцифровкой других текстов, входящих в канон, однако по-прежнему не размещённых в существующих коллекциях.

**1.2.2.** Графическое представление.

При представлении различных языков в электронном виде неизбежно возникает вопрос их отображения: далеко не у всех языков алфавит настолько прост для кодирования, как у большей части индоевропейских, для которых и начертания диакритик уже занесены в таблицы наиболее распространённых шрифтов.

Некоторые языки представляют особенно тяжёлый вызов для исследователя не одним-двумя символами, а целым собственным алфавитом, который был включён в Unicode только на рубеже тысячелетий [Gambäck, 2012, с. 79]. Это сыграло ключевую роль в развитии корпусной лингвистики, к примеру, амхарского языка [Gambäck, Asker, 2010], вернее, в её продолжительном отсутствии [Gambäck и др., 2009, с. 104]. Тем не менее с включением этих символов в Unicode проблема всё-таки была разрешена, и исследование данных языков начало развиваться значительными темпами; к данному моменту уже существует несколько достаточно крупных корпусов, для которых разрабатываются средства автоматической обработки текста [Singha и др., 2012, с. 31].

Для таких языков, как португальский, графическая нестандартность, напротив, в значительной степени играет исследователю на пользу. Позиционирование знаков препинания позволяет с большей ясностью разметить границы предложений, что в других языках, ввиду неоднозначности роли знаков препинания, было бы невозможно [Branco, Silva, 2003, с. 7].

Для большого числа языков проблемы представляет нестандартное обозначение диакритик. Такое достаточно часто происходит в исторических исследованиях даже достаточно широко представленных в корпусной лингвистике языков с большим количеством носителей, таких как испанский [Sánchez-Marco и др., 2011, с. 5]. Существует несколько способов постановки и разрешения этой проблемы.

В испанском из этих диакритик и размещённых перед ними букв приходится создавать графемы современного испанского алфавита, чтобы программы токенизации, тэггинга и разметки распознали требуемые слова, а не «неизвестные» до того комбинации стемов и окончаний [Sánchez-Marco и др., 2011, с. 5].

В арабском диакритики приходится, напротив, восстанавливать. Ряд исследователей предпочитают разрешать омонимию между словами с необозначенными диакритиками и без диакритик вообще при помощи ручной разметки [Alfaifi, Atwell C, 2013, С. 14 – 18]. Другие используют для этого специально обученную нейросеть [Ismail и др., 2014, с. 144]. Первый подход более характерен для корпусов небольшого размера, поскольку требует вложения значительного количества человеческих усилий и времени, второй – для корпусов большего объёма, поскольку именно достаточное количество единиц позволяет нейросети добиться относительно удовлетворительной эффективности.

В урду одному и тому же символу соответствует «несколько кодов Unicode одновременно», и исследователям приходится в процессе нормализации текста «заменять такие символы символами из других алфавитов урду» [Daud и др., 2017, с. 290]. Такая унификация представляется единственно возможным способом разрешения этой неоднозначности.

Наконец, для работы с рядом индийских языков приходится осуществлять конвертацию из символов самого языка в кодировке Unicode в заменяющие их в специально разработанной для этого системе обозначений символы ASCII [Reddy, Sharoff, 2011, с. 12 – 13], и обратно. Это трудоёмкая задача, требующая очень тщательной разработки системы правил, которая не будет создавать проблемы при обратном конвертировании, или хотя бы требовать минимального человеческого вмешательства при проверке получившегося в результате текста. В текущем состоянии развития компьютерной лингвистики, впрочем, обычно уже не требуется подобная процедура.

Старославянский язык концентрирует в себе практически все эти проблемы, и опыт решения каждой из них в той или иной степени оказывается полезен. Корпус Хельсинкского университета [CCMH], самый полный на данный момент, записан латиницей с использованием ASCII, активно ранее использовавшейся для машинной репрезентации старославянских текстов [Wátróbska, Kubiak, 2004], которую придётся подвергнуть конвертированию [Paliga, 2018]. Кириллица, используемая в старославянском, значительно больше по буквенному составу стандартной кириллицы, и особенно негативно это сказывается, когда требуется найти диакритику над редкой гласной. Некоторые особенно редкие старославянские символы не включены в стандарт Unicode 5.1 [Kempgen, 2008, с. 2 – 6].Сами диакритические знаки вторгаются в основу слова, принуждая исследователя удалять их из слов в процессе предобработки. При этом порой требуется и восстановить отсутствующий символ редуцированного, не внесённый, к примеру, при оцифровке. Наконец пунктуация в старославянском языке в функции разграничителя находилась в относительно зачаточной стадии своего развития, и при работе с ней требуется учитывать то, что она выполняет совершенно иные задачи, чем пунктуация в современных языках. Следовательно, опыт каждого из упомянутых выше исследований может быть полезным при создании графического представления корпуса старославянского языка.

**1.2.3.** Токенизация.

После того, как текст графически унифицирован, следующей задачей становится токенизация – «процесс отделения токенов друг от друга в поданном на вход тексте» [Deepali и др., 2018, с. 1607].

Обратим внимание на термин «токен». Традиционно под ним понимались «разделённые через пробел слова; единственными сложностями, да и то незначительными, становились пунктуация и форматирование текста» [Atwell, 2008, с. 509]. Однако это происходило в первую очередь из-за англоцентричности корпусной лингвистики как научной дисциплины, доминирования англоязычных корпусов в мире, которое продолжалось почти тридцать лет. В современной обстановке, когда на первый план выходят другие индоевропейские языки и неиндоевропейские языки, с их достаточно специфическим словообразованием, неоднозначным и с точки зрения лингвистов, и с точки зрения автоматической обработки, токенизация оказывается «обманчиво простой задачей» даже для таких близких английскому языков, как немецкий [Tufiş, 2009, с. 3]. Теперь токен скорее следует определить как «минимальная синтаксическая единица, например, слово, часть слова (или клитика), многословное выражение, либо же знак препинания» [Attia, 2007, с. 66].

Токенизация может осуществляться как вручную [Attia, 2007, с. 65], так и средствами автоматической обработки текста, собственно, токенизатором – программой, которая «конвертирует предложение в (…) токены» [Mohan Raj, Rajendran, 2017].

Существует несколько типов токенизаторов. В первом «токенизатор объединён с морфологическим анализом» [Attia, 2007, с. 67]. К преимуществам этого подхода следует отнести мультизадачность и большую эффективность: такая программа с лёгкостью различит слово, слово, соединённое с клитикой, либо же несколько слов, обладающих одной и той же синтаксической функцией. К недостаткам – то, что он может быть осуществлён либо вручную на не очень большом объёме текстов, либо при наличии достаточно крупного тренировочного корпуса и продвинутой нейронной сети, способной к выполнению такой задачи.

Второй тип – «простой токенизатор» [Attia, 2007, с. 67]. Это двухуровневая система, которая сначала проводит обычную токенизацию, поверх которой работает «детектор клитик» [Attia, 2007, с. 68]. Этот подход позволяет автоматизировать без значительных проблем достаточно крупную часть работы. Но несмотря на это, детектор клитик по-прежнему необходимо обучать. Более того, «тэггеру и парсеру приходится в итоге иметь дело с увеличившимся количеством неоднозначностей, становящихся результатом такой токенизации» [Attia, 2007, с. 68]. И тем не менее, исследователи считают этот подход наилучшим. Он позволяет разграничить модули программы и не нарушить принцип единственной ответственности для токенизатора [Attia, 2007, с. 68].

Последний способ токенизации – использовать её уже после проведения морфологического анализа. К преимуществам его следует отнести «более простой дебаггинг», к недостаткам – «недостаточную мощность» [Attia, 2007, с. 68]. Такой способ полагается, впрочем, исключительно на автоматическую разметку текста, предваряющую токенизацию, и применим только при автоматической же токенизации. Это не является недостатком, однако следует учитывать, что при ручной разметке корпуса преимущества такого подхода по меньшей мере сомнительны.

Как правило, исследователи не сосредотачивают в статье внимание на том, какое именно программное обеспечение используют в процессе. Встречаются упоминания класса Tokenizer языка программирования Java [Behera, 2017, с. 31]. Однако они практически не дают понимания того, какой именно из вышеупомянутых подходов был использован, поэтому оказываются практически бесполезными при попытке проектирования чего-то на их основе. К тому же, по всей видимости, речь идёт о стандартных классах Java, не адаптированных и с большим трудом адаптируемых (через наследование в пользовательский класс с перегрузкой всех методов) под нужды конкретного языка. Каждый следующий исследователь вынужден разрабатывать свой собственный токенизатор.

В ходе его создания требуется решать такие проблемы, как упомянутые выше графические сложности, создаваемые диакритическими значками, включаемыми в состав слов на правах отдельной лексемы [Sánchez-Marco и др., 2011, с. 5], либо же используемыми специфически знаками пунктуации [Branco, Silva, 2003, с. 7].

Наибольшие затруднения, впрочем, представляют собой поиск типов многословных выражений и создание соответствующих правил, которые подаются на вход токенизатора [Danso, Lamb, 2014, с. 2]. Под многословным выражением понимается «два или более слов, ведущих себя как одно слово синтаксически и семантически» [Attia, 2007, с. 68]. Для компьютера многословные выражения более удобно определить как те, для которых «стабильна дистанция между двумя лексическими токенами внутри текста (установленная через то, насколько мало стандартное отклонение такой дистанции)» и «статистически существенно количество раз, когда эти токены встречаются вместе (устанавливается через тест на логарифмическое правдоподобие дистанций)» [Tufiş, 2009, с. 5].

Выделяются три типа многословных выражений. К первому относятся «лексически, синтаксически и морфологически устойчивые единицы» [Attia, 2007, с. 68]. Второй включает в себя выражения, которые «могут варьироваться, однако сохранять строгую связь между компонентами» [Attia, 2007, с. 69]. Эти два типа «могут быть распознаны и отмечены токенизатором» [Attia, 2007, с. 69]. В отличие от них, третий «распознаётся только синтаксическим парсером» [Attia, 2007, с. 69]. Такие выражения характеризуются в первую очередь синтаксической свободой, которая и делает их особенно тяжёлой, практически невыполнимой, задачей для токенизатора, разрешить которую можно разве что с использованием встроенных словарей [Attia, 2007, с. 69].

Как именно поступать с подобными единицами в процессе токенизации, не совсем ясно. Ряд исследователей предпочитает передавать как отдельный токен каждое слово, а его вхождение в состав более крупной единицы отмечать уже непосредственно в процессе ручного редактирования результата [Крючкова, Гольдин, 2010, с. 13]. Другие же настаивают на том, чтобы программа сразу находила токены и выделяла их как отдельные единицы, несмотря на наличие внутри пробелов [Attia, 2007, с. 69].

Токенизация предстаёт перед нами сложным процессом, который на первый взгляд кажется тривиальным, однако в итоге таит в себе множество неочевидных для исследователя проблем, которые предстоит разрешать индивидуально, поскольку универсальные системы отличаются высокой степенью неточности при работе не с английским языком, а специализированных для многих языков просто на данный момент не существует.

**1.2.4.** Предобработка текста как многоуровневая задача.

Таким образом, прежде чем приступать к разметке и помещению в корпус текстов, исследователю предстоит осуществить значительный объём подготовительной работы.

В первую очередь, необходимо собрать тексты. Отобрать источники, распределить их по жанрам, возможно, даже устранить те, которые не подходят для решения изначальной задачи, в рамках которой и был собран корпус.

Затем необходимо провести графическую унификацию текстов, прежде всего для устранения возможных проблем в ходе токенизации, проводимой следом, а также для приведения текста в исходную форму, если он был предварительно конвертирован в ASCII для обработки.

Наконец тексты нужно разделить на токены, синтаксически элементарные единицы, чтобы можно было приступить к различным видам разметки текста, в частности, морфологическому анализу, базовому и первичному этапу работы с корпусом.

* 1. Частеречная разметка.

**1.3.1.** Общая характеристика частеречной разметки как лингвистической проблемы.

Частеречная разметка – «метод присваивания ярлыка грамматической категории каждому токену, основываясь на лингвистической и контекстуальной информации из предложения» [Behera, 2017, с. 18]. Формально её можно задать следующим образом: «При наличии последовательности слов *w1…wn* исследователь стремится к нахождению соответствующей ей последовательности тэгов *t*1…*tn*, выбранной из набора тэгов *T*» [Daundapat и др., 2007, с. 221]. К этому определению, вероятно, следует добавить то, что последовательность слов принадлежит набору слов языка данного текста *W*: словосочетание «язык данного текста» в данном случае используется для включения неизвестных слов основного языка текста, а также вероятных иноязычных вкраплений, и для тех случаев, когда язык текста в точности определить невозможно. Главной задачей, которую приходится решать исследователю, становится, соответственно, «разрешение неоднозначности» [Loftsson, 2008, с. 47].

История решения задачи частеречной разметки достаточно длинная, она берёт своё начало в ту же эпоху, когда зародились первые крупные языковые корпусы, такие как Брауновский, и развивается и по сей день. Она прошла долгий путь от ручного, совершаемого десятками исследователей, труда, до относительно быстро исполняемой, но от того не менее трудоёмкой задачи, для которой используются новейшие разработки в области машинного обучения и достижения теоретической математики, такие как скрытые марковские модели, усиленные алгоритмом Витерби [Rajendran, Krishnakumar, 2019, с. 132 – 136].

Будучи одной из первых задач, которую пришлось разрешать на объёмах целых корпусов, снабжая исследователя достаточно ценной информацией, частеречная разметка оказалась источником важных вспомогательных данных при решении целого ряда других задач, начиная от разметки синтаксической и заканчивая извлечением информации [Reddy, Sharoff, 2011, с. 11]. В частности, в качестве возможного применения называется использование при составлении словарей [Abumalloh и др., 2016, с. 45].

Алгоритм разрешения задачи частеречной разметки может быть следующим [Singha и др., 2009, с. 33]:

«Шаг 1: Ввести текст на естественном языке.

Шаг 2: Повторять шаги 3 – 7, пока весь текст не будет введён и проанализирован.

Шаг 3: Токенизировать текст, поданный на вход системы.

Шаг 4: Если в слове присутствует словообразовательная или словообразующая морфема, либо более одного корня, то слово следует передать стеммеру для разделения.

Шаг 5: Морфологический анализатор проверяет слово на предмет соответствия в лексиконе.

Шаг 6: Если соответствие найдено, слову присваивается нужный тэг.

Шаг 7: Если для одного слова найдено несколько тэгов, то тэггер принимает решение, основываясь на правилах.

Шаг 8: Возвращается текст с присвоенными тэгами.

Шаг 9: Из возвращённого текста извлекаются неизвестные ранее слова.

Шаг 10: Создаётся новая запись для неизвестного нового слова в лексикон.

Шаг 11: Словам в новой записи присваиваются новые правила».

**1.3.2.** Набор тэгов: самостоятельная разработка или заимствование?

Из представленного выше механизма становится очевидным, что перед тем, как приступить к частеречной разметке, исследователь сталкивается с необходимостью разработки набора тэгов, представляющего собой «категории тэгов, которые могут быть использованы для присвоения тэга каждому слову, основываясь на контексте» [Kumar S. и др., 2016, с. 8017]. Каждый тэг представляет собой значение определённой части речи, единицы конечного множества классов слов, определённых исследователем в соответствии с релевантными в рамках корпуса критериями. Эти критерии могут существенно разниться от исследования к исследованию, поскольку «дискуссия о частях речи, о классификационных критериях для их разграничения, об установлении границ между частями речи и пути определения этих границ по-прежнему продолжается, и продолжится в будущем» [Koseska-Toszewa, Roszko, 2008, с. 80]. Подход к созданию тэгсета у большей части исследователей разнился в соответствии с задачами, которые тем приходилось решать, однако всё же «общим оставалось стремление обогатить языковые корпусы лингвистическим анализом, чтобы повысить их полезность в большом спектре задач по обработке естественного языка» [Zeroual и др., 2017, с. 173]. Совокупности тэгов, которые изначально присвоены определённому слову в тренировочном корпусе, определяются как тэг-профиль этого слова [Loftsson и др., 2009, с. 106]. Наборы тэгов, как правило, достаточно обширны, поэтому зачастую для снижения нагрузки на машину, выполняющую автоматическую обработку, исследователю приходится использовать целый ряд методик. Первая из них – модификация набора тэгов, его «упрощение и обновление тренировочного корпуса с целью того, чтобы тот отражал разметку текста новым набором тэгов» [Loftsson и др., 2009, с. 107]. В другом случае используется создание словаря тэгов, в случае с которым «при сопоставлении тэга *t1* на выходе тэггера с тэгом *t2*золотого стандарта, тэги *t1* и *t2* приравниваются к упрощённым тэгам *m1* и *m2* , которые и сопоставляются между собой» [Loftsson и др., 2009, с. 107].

Вопрос о составлении набора тэгов особенно остро стоит в лингвистике в последние годы. Стандартом считается тэгсет EAGLES [Rajendran, Krishnakumar, 2019, с. 147]. Однако он разработан для представленных наиболее активно в корпусах языков, по большей части индоевропейских. Более того, ряд исследователей утверждают, что конкретно эта отрасль лингвистических разработок испытала слишком большое влияние английской корпусной лингвистики и английского языка, что сделало многие наборы тэгов неприменимыми для языков, морфология которых значительно отличается от морфологии английского [Zeroual и др., 2017, с. 171]. Это привело к разработке новых наборов тэгов, приспособленных под нужды конкретных языков, таких как арабский [Zeroual и др., 2017, с. 174], или же манипури [Singha и др., 2012, с. 32].

Дальнейшее развитие лингвистики привело к необходимости совершенствования существующих наборов тэгов даже в языках, изначально вполне обеспечиваемых уже существующими. Так, особое положение имён собственных как категории привело к необходимости включения в состав набора тэгов специальных обозначений для них, а программы разметки потребовалось оснастить механизмами распознавания именованных сущностей [Loftsson и др., 2009, с. 108]. Помимо этого, ряд исследователей настаивает на особом способе присвоения тэгов лексемам с семантикой модальности, что выглядит вполне логично, учитывая особенности их словоизменения и синтаксической роли [Netzer и др., 2007, с. 64].

Все эти проблемы остаются неразрешёнными, оставленными индивидуально каждому отдельному исследователю, создающему набор тэгов.

**1.3.3.** Тэггеры: характеристика текущего состояния.

Под тэггером в данной работе мы подразумеваем некий механизм, выполняющий задачу частеречной разметки, а именно присваивания некоему набору слов некоего набора тэгов. Следует заметить, что в целом работа, которую решает тэггер, может быть с вполне удовлетворительной эффективностью выполнена и человеком (с пределом в 97% [Gambäck и др., 2009, с. 105]), однако на больших объёмах текстов затраты, временные и финансовые, оказываются слишком серьёзными, чтобы не обратиться к помощи машины.

Тем не менее, и сами машины далеко не совершенны. Вне зависимости от того, как классифицируется конкретная система, большие данные в неё приходится подавать частями, в противном случае, по мнению ряда исследователей, возникает исключение превышения допустимого объёма памяти [Kumar S. и др., 2016, с. 8019]. Однако эта задача в целом разрешима программными средствами, и к проблемам именно лингвистического анализа, пожалуй, относится в наименьшей степени.

Особое внимание при составлении любой системы следует уделить частеречным категориям, особенно сложным для интерпретации, а именно «возвратным местоимениям, указательным местоимениям, герундийным формам, нефинитным и смысловым глаголам» [Behera, 2017, с. 27]. При построении системы на правилах раздел, посвящённый таким словам, следует разработать особенно тщательно, а при разработке моделей машинного обучения – постараться указать на них машине при помощи механизмов внимания.

Общими являются и метрики, используемые при анализе показателей. К их числу стоит отнести, в первую очередь, классические, такие как общая точность (количество слов, тэг которым был присвоен корректно, разделённое на общее количество слов [Martinez и др., 2018, с. 891]), полнота (количество слов, тэг которым был присвоен корректно, разделённое на количество слов, которым этот тэг следовало присвоить) точность (количество слов, тэг которым был присвоен корректно, разделённое на общее количество слов, которым он был присвоен), и F-мера (усреднённое между двумя последними метриками) [Daud и др., 2017, с. 295], Ряд авторов прибегает к более общим методам, таким как средняя квадратичная ошибка и нормализованная средняя квадратичная ошибка [Yousif, 2013, с. 44].

Соответственно, машинные средства частеречной разметки делятся по целому ряду оснований.

Например, они могут быть основаны на правилах, разработанных лингвистами, либо же использовать статистические методы, включая методы машинного обучения, для самостоятельного выведения правил, основываясь на предоставленных объёмах данных [Loftsson, 2008, с. 48].

Системы, основанные на правилах, начали разрабатываться в начале второй половины двадцатого века [Abumalloh и др., 2016, с. 47]. Они используют «разработанные вручную правила, чтобы разметить корпус» [Daud и др., 2017, с. 45].

Проблемы систем, основанных на правилах, сводятся к тяжёлой разработке непосредственно системы правил. Это менее затратно по человеко-часам, чем вручную размечать корпус, однако в условиях ограниченности времени исследователя годом или двумя, по словам учёных, уже разрабатывавших подобные системы, тоже не представляется возможным [Danso, Lamb, 2014, с. 1].

Другие исследователи, впрочем, утверждают, что для некоторых языков вполне достаточно нескольких месяцев [Loftsson, 2008, с. 49]. Тем не менее, это всё равно требует значительных усилий. Системы, разрабатываемые этими исследователями, к примеру, оснащены модулями распознавания неизвестных слов [Loftsson, 2008, с. 59] и идиом [Loftsson, 2008, с. 60]. Помимо этого, разрабатываются контекстуальные правила [Loftsson, 2008, с. 60] и глобальные эвристики, которые «при разрешении неоднозначности для конкретного слова позволяют обращаться к слову, не находящемуся в пределах технического окна» [Loftsson, 2008, с. 61]. Наконец, как и в случае с модальными лексемами, для ряда глаголов пришлось разработать отдельный набор правил [Loftsson, 2008, с. 62].

К возможным недостаткам систем, работающих на данных, в свою очередь, относится необходимость наличия тренировочного корпуса (а значит, затрат человеческих усилий на его ручную разметку) достаточно больших объёмов, чтобы на его материале можно было обучить нейронную сеть, простую модель машинного обучения, или же вывести элементарную статистическую закономерность [Saharia и др., 2009, с. 33; Mirzanezhad, Feizi-Derakhshi, 2016, с. 1099].

Исходя из метода решения этой проблемы, выделяются системы, обучающиеся с учителем, то есть, такие системы, которые требуют предварительно размеченного корпуса данных [Rajendran, Krishnakumar, 2019, с. 136], и системы, обучающиеся без учителя, способные приспособиться к новой системе сами [Gambäck и др., 2009, с. 106]. И те, и другие, впрочем, требуют больших объёмов исходного текста.

Системы, работающие на больших данных, в свою очередь подразделяются на простые статистические модели [Behera, 2017, с. 20] и различные модели с использованием машинного обучения, среди которых отдельно выделяются модели, работающие на нейросетях [Abumalloh и др., 2016, с. 49]. Последние на данный момент набирают всё большую популярность, однако более примитивные модели машинного обучения до сих пор показывают достаточно высокую эффективность. Следует заметить, что терминологически сфера развилась ещё не достаточно сильно, из-за чего, например, одна и та же система может получить название статистической и стохастической без существенных различий в архитектуре [Behera, 2017, с. 21].

Выделяют несколько возможных типов моделей машинного обучения.

Одним из наиболее редких является SVM [Behera, 2017, с. 20]. Это «метод машинного обучения с учителем, в котором данные на входе представляются как векторы в многомерном пространстве, а SVM находят гиперплоскость, (…) разделяющую пространство ввода на два через максимизацию разрыва между положительными и отрицательными вхождениями среди данных» [Gambäck и др., 2009, с. 105].

Чаще встречаются модели условных случайных полей, «метод статистического моделирования, используемый для машинного обучения в приложениях, распознающих различные паттерны» [Kumar S., 2016, с. 8018]. Формализовать его можно следующим образом [Behera, 2017, с. 24]: «При *G* – графе показателей на *y*, *p(y|x) –* условное случайное поле, если обобщённая функция раскладывается на множители согласно *G* для любого зафиксированного *x*:

Модель условных случайных полей создавалась как альтернатива, но так и не смогла вытеснить скрытые марковские модели, представленные сейчас в тэггерах во всём многообразии форм [Dandapat и др., 2007, с. 222].

Скрытая марковская модель основана на принципе конечного автомата [Jurish, 2003, с. 5], «кортежа из пяти элементов *A = (∑, Q, q0, δ, F),* где *∑ –* конечный алфавит, Q – конечный набор переходов между состояниями автомата, q0  ∈ Q – начальное состояние автомата, – формула перехода, F ⊆ Q – набор конечных состояний». В свою очередь, скрытая марковская модель определяется как такой же кортеж, где алфавит ∑ становится «алфавитом наблюдений», функция перехода определяется как «вероятностное распределение переходов между состояниями с выделенным переходом между состояниями *q1* и *q2* *A*(q1*, q2*)», а набор конечных символов определён как «вероятностное распределение соответствий символов алфавита состояниям, , где *B(q, a)* – вероятность наблюдения символа *a* при состоянии *q*» [Jurish, 2003, с. 8]. Нетрудно догадаться, что символы в данном случае – токены, а состояния – тэги.

Скрытая марковская модель используется для частеречной разметки сравнительно давно, и со временем был разработан целый ряд способов улучшить её функционирование. В частности, практически с самого начала применялся алгоритм Витерби, «метод динамического программирования, который эффективно вычисляет для скрытой марковской модели и данной последовательности наблюдений *o1..n* ∈ ∑n  последовательность символов *q1..n* ∈ Qn, которую с наибольшей вероятностью породит последовательный ввод *o1..n* при определённых параметрах модели» [Jurish, 2003, с. 8]. Ещё одним математическим методом стал метод максимальной энтропии [Dandapat и др., 2017, с. 222]. Были попытки усилить модель и переносом обучения: её сначала обучали на языке с большим количеством ресурсов, а затем подкрепляли куда меньшим набором данных на целевом языке [Reddy, Sharoff, 2011, с. 17].

Отдельно от других методов стоят гибридные, сочетающие в себе, к примеру, статистическую систему и систему на правилах [Mohamed Elhadj, 2009, с. 795]. Эти системы могут работать параллельно, а могут группироваться в ансамбли [Danso, Lamb, 2014, с. 3]. Совместность их работы обеспечивается либо голосованием (при котором у результатов разных моделей может быть либо разный, либо одинаковый вес, и решение принимается после сопоставления этих результатов), либо соединением (последовательным запуском моделей, где каждая следующая разрешает всё более частные противоречия) [Gambäck и др., 2009, с. 106].

Все вышеуказанные методы нашли своё отражение в разработанных исследователями тэггерах. Статистические модели легли в основу тэггера английского языка CLAWS [Rajendran, Krishnakumar, 2019, с. 146] и тэггера арабского языка TreeTagger [Kumar S. и др., 2016, с. 8019]. SVM используется в SVMTool [Gambäck и др., 2009, с. 106], CRF – в CRF++ [Behera, 2017, с. 31]. Скрытая марковская модель применена в тэггере TnT [Reddy, Sharoff, 2011, с. 12], а её усиленная методом максимальной энтропии версия – в утилите MALLET [Gambäck и др., 2009, с. 106]. Различные модели могут быть использованы во фреймворке EPIC языка Scala [Kumar S. и др., 2016, с. 8019].

**1.3.4.** Частеречная разметка как сфера поисков исследователя.

Проанализировав всё выше написанное, следует отметить, что частеречная разметка и создание набора тэгов – индивидуальная для каждого языка задача. При наличии достаточного количества времени и человеческих ресурсов предпочтение следует отдать ручной разметке, при их относительном недостатке – тэггеру, работающему на правилах. В случае же работы с корпусами любых размеров в условиях нехватки времени лучше всего показывают свою эффективность скрытые марковские модели, усиленные математическими или лингвистическими средствами. Наконец, в случае слишком серьёзных затруднений при работе единственной системы возможно собрать ансамбль, который станет менее интерпретируемым, но при этом более эффективным.

* 1. Лемматизация.

**1.4.1.** Определение лемматизации и основные области её применения в корпусной лингвистике.

Лемматизация – процесс приведения слова к его лемме или основной форме. Этот процесс схож со стеммингом, однако стемминг представляет собой более простой алгоритм, который обычно определяет только первые *x* графем, общие для всех форм слова, в то время как лемматизация сопоставляет общую форму данному набору словоформ. Например, не существует способа провести стемминг для старославянских *мене* ‘я-GEN’ и *азъ* ‘я-NOM’, в то время как леммой обоих является *азъ* ‘я’.

Лемматизация широко применяется в лингвистике, в частности, при создании корпусов [Hardie и др., 2014; Schryver, Nabierye, 2018; Camps и др., 2020]. Корпусы, токены которых были лемматизированы, легче подвергаются анализу. Поиск по ним также упрощается. Наконец, при помощи таких корпусов значительно легче составлять словари.

**1.4.2.** Лемматизация как задача автоматической обработки естественного языка.

Лемматизация находилась в сфере интересов автоматической обработки естественного языка последние несколько десятилетий [Hann, 1974]. В рамках данной области лемматизация может быть определена несколькими способами, в зависимости от конкретного подхода, который использует исследователь.

Первое из определений базируется на более старом подходе, представленном в работах, в рамках которых лемматизация рассматривается как гибридная задача из трёх шагов [Mills, 1998; Chrupała, 2006; Plisson и др., 2008; Gesmundo, Samardžić, 2012; Radziszewski, 2013]. В первую очередь исследователю необходимо описать правила, по которым слова в определённом языке лемматизируются. Затем модель обучается для решения задачи мультиклассовой классификации. Эта задача состоит в соединении лемматизационной модели и словоформы. Инструмент, решающий данную задачу, может быть как алгоритмом градиентного бустинга, таким как XGBoost [Tianqi, Carlos, 2016], так и статистической моделью, например, скрытой марковской (СММ) [Jiampojarman et al., 2008]. На третьем этапе уже определённое правило применяется к слову, и результат этого действия, а именно определённая программой лемма, подаётся на выход.

Второе определение возникло благодаря новому подходу к лемматизации, активно используемому в последнее десятилетие, превращающему его в процесс, состоящий лишь из одного действия [Kanerva и др., 2020]. На вход программа принимает словоформу и вспомогательную информацию, такие как часть речи, грамматические значения слова, левый контекст, на выход подаёт лемму. Обычно это достигается через декодинг информации на входе, преобразования полученного тензора, и энкодинг последнего в информацию, подаваемую на выход. Выбор инструментов при этом подходе более ограничен. Модели энкодер-декодер обычно создаются при помощи нейросетей с достаточно сложной архитектурой [Ljubešić, Dobrovoljc, 2019]. Предпочтительной архитектурой является модель «последовательность-в-последовательность» (seq2seq), которая принимает на входе слово и опциональную вспомогательную информацию, а на выходе выдаёт лемму [Bergmanis, Goldwater, 2018]. В этом случае автоматическая лемматизация может быть определена как превращение нейронной сетью последовательности, состоящей из слова и лингвистической информации о нём, в последовательность, которая состоит из леммы.

Лемматизаторы разделяются на универсальные [Cartoni, 2009; Straka, Straková, 2017; Bergmanis, Goldwater, 2018; Kanerva и др., 2020] и специализированные, предназначенные для выполнения задачи на материале конкретного языка [Džeroski, Erjavec, 2000; Groenewald, 2007; Tamburini, 2013] или конкретной части речи [Prinsloo, 2012; Gouws, Prinsloo, 2012; Nthambeleni and Musehane, 2014], или группы слов внутри конкретной части речи [Fernández, 2020], или класса слов с очень специфическим синтаксическим поведением, таком как слова внутри устойчивых выражений [Farkas и др., 2008; Mulhall, 2008; Kosch, 2016]. В некоторых работах лемматизатор и тэггер объединяются в цельную модель [Spyns, 1996; Aduriz и др., 1998]. Некоторые лемматизаторы, разработанные для одного из языков, могут быть дополнительно использованы на материале других [Groenewald, 2009].

Автоматическая лемматизация изначально осуществлялась системами, основанными на правилах [Evans, 2006; Jursic, 2010]. Затем произошёл сдвиг в сторону моделей машинного обучения, основанных на статистике [Mzamo и др., 2015]. Наиболее успешные современные модели универсальны [Straka, Straková, 2017; Bergmanis, Goldwater, 2018; Kanerva и др., 2020]. Для обучения, валидации и тестирования этих моделей используются корпусы UD [Straka, Straková, 2017], а сами эти модели становятся частью общего пайплайна UD [Straka и др., 2016].

**1.4.3.** Лемматизаторы старославянского языка.

Интерес к лемматизации старославянских текстов рос на протяжении последних пяти лет. В используемых моделях, как правило, применялась архитектура seq2seq с механизмом внимания [Sutskever и др., 2014; Cho и др., 2014]. Результаты с момента появления первых публикаций были в той или иной степени удовлетворительными [Podtergera, 2016]. Однако в случае с определённой частью старославянского канона, *Codex Marianus*, который является частью коллекции UD, эти результаты были превзойдены универсальными лемматизаторами, которые постепенно достигли уровня общей точности в 95–97% [Straka, Straková, 2017; Bergmanis, Goldwater, 2018; Kanerva и др., 2020]. Эти модели в основном основаны на UD. Важно заметить, что старославянский набор данных в рамках UD не включает знаки препинания, фрагменты и цифры. Возможным решением может стать использованием гибридной модели, как это уже было сделано для древневосточнославянского и старорусского языков [Berdičevskis, 2016].

**1.4.4.** Лемматизация при создании корпуса старославянского языка.

Таким образом, подходы к лемматизации проделали путь от систем на правилах и статистических методов до рекуррентных нейросетей. Лемматизации в данный момент возможно подвергнуть токены любого языка, в том числе старославянского, при наличии текстов достаточного объёма.

Тем не менее, в лемматизации старославянского языка решены не все задачи. Исследователям предстоит выявить способ улучшения существующих систем в направлении повышения эффективности взаимодействия с фрагментарными токенами, цифрами и знаками пунктуации.

* 1. Выводы.

Таким образом, восстановлен порядок создания корпуса языка, так или иначе отличающегося от языков, традиционно выступавших объектами изучения корпусной лингвистики, таких как английский.

В первую очередь, подчёркнуты направления развития современной корпусной лингвистики, определены её центры: Аравия, Индия, субсахарская Африка и Южная Америка. Проанализированы методологии, используемые для разработки корпусов этих территорий, а также их влияние на текущие тенденции в дисциплине в целом.

Затем значительное внимание было уделено вопросам подготовки корпуса, в частности, подбору текстов, их графической предобработке и токенизации. Были выявлены проблемы, с которыми предстоит столкнуться в ходе реализации практической части, в частности, особенности транслитерации существующих корпусов старославянского языка.

Изучен вопрос современных подходов к частеречной разметке. Наиболее распространёнными её методами являются подход, основанный на правилах, и подход, связанный с использованием скрытых марковских моделей, а также их различных усовершенствований. Выявлены ключевые метрики, по которым происходит оценка тэггеров. Исследован вопрос создания набора тэгов.

Наконец, выявлены текущие тенденции в лемматизации: универсализация и использование seq2seq-моделей вместо статистических методов и систем на правилах. Лемматизаторы старославянского языка за последние пять лет появились и активно развились, на некоторых наборах данных достигнув точности в 97%.

Практически в каждом направлении было выяснено, что подход к решению поставленной проблемы индивидуален для конкретного языка, а следовательно, при разработке корпуса старославянского языка исследователь будет вынужден опираться на существующий опыт, действуя при этом совершенно самостоятельно.

## **Глава 2. Базовая теоретическая характеристика языка текстов старославянского канона.**

**2.1.** Старославянский язык: определение, особенности, периодизация.

Старославянский язык характеризуется исследователями как относящийся к «южнославянскому типу, точнее – к македонскому, очень близкому к типу болгарских говоров» [Мейе А., 1934, с. 7]. Его считают очень близким к общеславянскому языку на завершающем этапе существования последнего [Мейе А., 1934, с. 7], однако при этом активно указывается на то, что он представлял собой восточноюжнославянский диалект, и не мог быть предковой формой для славянских языков, возникших позднее [Kamphuis J., 2020, с. 5]. Из известных текстов, по мнению исследователей, ни один «не передаёт вполне точно оригинала» [Мейе А., 1934, с. 8]. Утверждается, что каждая рукопись «по существу отражает влияние какого-то языка», и известны в основном те рукописи, которые отражают влияние южнославянских языков [Мейе А., 1934, с. 8]. Также следует заметить, что каждый из памятников старославянского языка носит черты диалектного влияния, в особенности влияния диалекта переписчиков [Мейе А., 1934, с. 8].

Дискуссионным в вопросах генеалогической преемственности является разве что вопрос о церковнославянских языках, в ветви которых старославянский чаще всего называют староцерковнославянским языком (термин, который отвергался ранее [Вайан А., 1952, с. 15], но всё чаще принимается в современных исследованиях [Kamphuis J., 2020, с. 4]). Однако даже в этом случае возникает вопрос о непосредственной предковости староцерковнославянского языка по отношению к другим церковнославянским языкам.

В каком отношении состоят отдельные традиции текстов? Возможно, все они в той или иной степени восходят к традиции первой родины (о периодизации – см. далее) [Kamphuis J., 2020, с. 5], текстам, написанным самими Кириллом и Мефодием. Однако в силу того, что этих текстов не сохранилось вообще [Селищев А. М., 1951, с. 67], предположение носит лишь гипотетический характер. Тексты же даже непосредственно второй и третьей родин [Kamphuis J., 2020, с. 5] значительно отличаются друг от друга лингвистически. При этом перед нами не различия формы-предка и формы-потомка, дифференциация проходит по разным рефлексам одного и того же явления [Kamphuis J., 2020, с. 6]. Вопрос исходного лингвистического единства традиций остаётся на данный момент неразрешённым, однако, так или иначе, все они исторически составляют единый канон.

Старославянский язык отделяется от других славянских языков, равно как и от иных церковнославянских традиций «рядом географических и темпоральных демаркационных линий» [Kamphuis J., 2020, с. 4]. При этом, как упоминалось выше, внутри самой старославянской традиции существует некоторая степень вариативности, как исторической, так и лингвистической [Kamphuis J., 2020, с. 4]. Однако вместе с тем исследователи выделяют ряд особенностей, по которым старославянский язык отделяется от всех остальных славянских языков и диалектов своего времени.

Ключевой особенностью в этом ряду является особенность восточноюжнославянских диалектов, а именно «рефлексы \**tj* и \**dj*, *št* и *žd*  соответственно» [Kamphuis J., 2020, с. 6]. Следовательно, «ни один текст, в котором зафиксированы иные рефлексы \**tj* и \**dj*, не может являться частью старославянского канона; исключением из этого правила традиционно считаются Киевские листки, демонстрирующие смешение южнославянских и западнославянских черт» [Kamphuis J., 2020, с. 6].

Помимо этого, указывается на «ряд черт, позволяющих идентифицировать старославянский язык как южнославянский, однако они не эксклюзивны, что значит: существуют при этом или в западных, или в восточных славянских языках» [Kamphuis J., 2020, с. 7]. К этим чертам относится, к примеру, рефлекс *l* праславянских групп согласных *\*tl* и *\*dl*, характерный также и для восточнославянских языков. К нему примыкает рефлекс RaC сочетания гласного с восходящим ударением с плавным перед согласным \*óRC, также фиксируемый в диалектах словацкого [Kamphuis J., 2020, с. 7].

Помимо этого, от текстов различных изводов церковнославянского языка старославянский отличают разное отражение процессов исчезновения носовых и редуцированных гласных [Kamphuis J., 2020, с. 7]. В процессе перехода от старославянского языка к изводам исчезает этимологическое написание соответствующих графем там, где изначально существовали носовые гласные [Kamphuis J., 2020, с. 7]. Помимо этого, следствием падения редуцированных, происходившим уже в старославянском, становится появление одноеровой орфографии, характерной для более поздних памятников [Kamphuis J., 2020, с. 7].

Старославянский язык показывает значительную степень вариативности даже без учёта того, что не все тексты канона удовлетворяют критерию бытия текстами языка, что объясняется наличием двух центров книжности и периодом существования в 200 лет [Kamphuis J., 2020, с. 8].

На фонологическом уровне ключевым различием между Охридской и Преславской школами является отражение процесса падения редуцированных [Kamphuis J., 2020, с. 8]. Так, в древних памятниках, например, Зографском евангелии, падение редуцированных, по всей видимости, не отражено, во всяком случае, в аспекте вокализации редуцированного заднего ряда *ъ* в *о* [Kamphuis J., 2020, с. 8]. В текстах Преславской школы *ъ* пишется в сильной позиции, по всей видимости, маркируя произношение звука *шва*, рефлекса редуцированного заднего ряда в сильной позиции в современном болгарском [Kamphuis J., 2020, с. 8]. В текстах Охридской школы отражается процесс вокализации редуцированного заднего ряда в *о* [Kamphuis J., 2020, с. 8].

Существует также и лексическое различие между двумя традициями: в Охридской школе предпочтение отдавалось славянским по происхождению словам, в то время как в Преславской школе традиционно использовались грецизмы [Kamphuis J., 2020, с. 7]. Тем не менее этот признак наименее устойчивый, поскольку синонимические лексические единицы различного происхождения встречаются в текстах обеих школ [Kamphuis J., 2020, с. 7].

Длительный период существования старославянского языка делится на «три или четыре периода, ограниченные темпорально и географически, также называемые *Heimat* ‘родина’» [Kamphuis J., 2020, с. 5].

Нам неизвестны тексты периода первой родины, написанные непосредственно Кириллом. Существуют предположения о том, что они могли «отражать македонский диалект, известный братьям [Кириллу и Мефодию], жившим в Фессалониках» [Kamphuis J., 2020, с. 5], однако пока не найдено ни одной рукописи, утверждать это с точностью не представляется возможным.

Вторая родина – это период моравской миссии, следующих двадцати лет после того, как Кирилл и Мефодий покинули Константинополь [Kamphuis J., 2020, с. 5]. Вопрос того, известны ли какие-то тексты этого периода, остаётстся открытым: Киевские листки, очевидно, связаны со второй родиной, и традиционно к ней причисляются в силу того, что были написаны глаголіцей и лингвистически представляли собой смесь западнославянских и южнославянских диалектов [Kamphuis J., 2020, с. 5]. Однако непосредственно текстов периода второй родины на данный момент обнаружено не было.

Болгаромакедонский период третьей родины наступил после 885 года, даты окончания моравской миссии, и продолжался до конца десятого века [Kamphuis J., 2020, с. 5]. Не известны ни лингвистически схожие со старославянскими славянские тексты этого периода, ни тем более тексты непосредственно старославянского языка, написанные в рамках кирилломефодиевской традиции.

Наиболее многочисленны тексты четвёртого периода (или же «второй части периода третьей родины»), времени формирования и существования охридской и преславской традиций в Македонии и Болгарии соответственно. Эти тексты были написаны в одиннадцатом веке [Kamphuis J., 2020, с. 5].

Написанные в дальнейшем тексты, относящиеся к двенадцатому веку, и рядом исследователей называемые «поздним старославянским», практически сливаются с изводами церковнославянского языка на территориях расселения восточных, южных и западных славян [Kamphuis J., 2020, с. 6]. Здесь граница между старославянским как языком-предком и церковнославянскими языками как языками-потомками проходит особенно близко, и установить точно, с каким именно из языков сталкивается исследователь, оказывается особенно затруднительно. Традиционно большая часть этих текстов не включается в канон текстов старославянского языка, за исключением тех случаев, когда на это существуют объективные лингвистические причины, как, к примеру, они существуют для Енинского апостола [Kamphuis J., 2020, с. 6].

Таким образом, старославянский язык – восточноюжнославянский язык, не являющийся предковым ни для одного современного славянского языка, за исключением, вероятно, только различных изводов церковнославянского языка. Уникальным отличием старославянского языка от последующих изводов церковнославянского языка является то, что он, по всей видимости, частично фиксировал живой диалектный материал; на языке, схожем со старославянским, действительно говорили. Однако сам старославянский язык при этом был исключительно письменным. Тексты старославянского языка характеризуются значительной степенью гетерогенности, графической, исторической и лингвистической [Miklas, Sadovski, 2014]. Несмотря на наличие достаточно чётких критериев определения принадлежности текста к старославянскому канону, ряд исследователей предпочитает уделять внимание другим чертам в случае с рядом рукописей [Мейе А., 1934, с. 9]. Один документ причисляется к канону на основании совокупности второстепенных черт и, скорее, исторических причин, чем непосредственно лингвистических. Помимо этого, длительный период существования языка наложил отпечаток на вариативность внутри даже наиболее гомогенных частей его массива, в особенности в области отображения гласных фонем на письме. Следствием большого времени существования и постоянной смены Кириллом, Мефодием и их учениками места деятельности, стала периодизация истории старославянского языка. Первый (~862/863 [Kamphuis J., 2020, с. 5]), второй периоды, а также первая часть третьего, носят сугубо исторический характер, нам неизвестны тексты этого времени (за исключением всё тех же Киевских листков, чей статус в качестве памятника кирилломефодиевской традиции всё же спорен). С двенадцатого же века влияние местных языков на старославянский приводит к формированию церковнославянских изводов. Поэтому канон старославянского языка в основном состоит из второй части периода третьей родины, текстов конца десятого – одиннадцатого веков, из которых же исследователю и стоит составлять будущий корпус.

**2.2.** Графические системы, репрезентирующие тексты канона старославянского языка.

Тексты канона старославянского языка в данный момент в различных источниках представлены тремя различными системами письменности: глаголицей, кириллицей и латиницей.

Вопрос о том, какая из первых двух систем письменности являлась исконной для старославянского языка, оставался открытым на протяжении девятнадцатого века, когда были открыты первые старославянские рукописи. Первоначально доминирующей выступала точка зрения, согласно которой кириллица была применена для написания старославянских рукописей раньше [Срезневский И. И., 1866]. Однако постепенно, с открытием всё более древних глаголических рукописей, входивших по мере их обнаружения в научный оборот и пополнявших собой старославянский канон, всё больше исследователей выдвигало тезис о том, что первичным алфавитом была глаголица [Шафарик Й., 1860]. К середине двадцатого века данное мнение окончательно закрепилось в славистике [Селищев А. М., 1951] [Вайан А., 1952], при этом вопрос о том, какой из алфавитов первичен, оставался неразрешённым. Однако тот факт, что найденные глаголические рукописи древнее найденных кириллических, на данный момент оставляет гипотезу о спонтанном, независимом, более раннем происхождении кириллицы по меньшей мере спорной.

Вопрос происхождения глаголицы в данный момент остаётся неразрешённым, различные попытки возвести её к иным письменным системам потерпели неудачу [Гранстрем Е. Э., 1955]. Вместе с тем, действительно странным выглядит ряд попыток обозначить глаголицу как исконно древнерусское письмо [Там же, с. 302]. По всей видимости, глаголица действительно «была изобретена Кириллом» [Там же, с. 302] и была основана на «своеобразных и малоизвестных… видах греческого письма» [Там же, с. 303].

Глаголица содержит 39 графем [Гранстрем Е.Э., 1955, с. 305], большая часть из которых соответствует конкретным фонемам старославянского языка. Оставшиеся могут обозначать различные звукоизменения и звукосочетания. К их числу относятся йотированные графемы, обозначающие либо продвинутый по ряду звук в позиции после мягкого согласного (ⰒⰓⰊⰅⰏⰎⰩⰜⰅ‘принимая’ [TITUS]), либо комбинацию йота и продвинутого по ряду звука в позиции после гласного (ⰏⰊⰎ*ⰑⰔⰕ*ⰊⰩ‘милостью’ [TITUS]). Некоторые буквы, такие как *иже*, зачастую дублируются, функционируя преимущественно в составе числовых обозначений [Гранстрем Е.Э., 1955, с. 305]. Сложный шипящий [ш’т’] может обозначаться сочетанием ⰞⰕ (ⰀⰞⰕ Ⰵ’если’ [TITUS]), или лигатурой Ⱋ (ⰀⰛⰅ’если’). Наконец, ряд графем функционирует только в заимствованиях, обозначая неславянские звуки или звукосочетания, к примеру, Ⱇ [ф] в ⰗⰀⰓⰊ*ⰔⰡ*Ⰺ ‘фарисеи’ [TITUS].

Глаголицей написана значительная часть древнейших старославянских текстов, в том числе: единственный текст времён второй родины, Киевские листки, один из первых найденных текстов, Клоцов сборник, Зографское и Ассеманиево Евангелия, ряд памятников, обнаруженных на Синае. Тем не менее эта система достаточно быстро перестаёт использоваться на значительной части *Slavia Orthodoxa* и некоторой части *Slavia Romana*, уступая место кириллице практически везде, за исключением территории современной Хорватии, где изменяется из округлой старославянской в угловатую [Афанасьева и др., 2016].

Глаголица была включена в стандарт Unicode, начиная с версии 4.0 [Unicode Standard]. Следовательно, на большинстве современных устройств, во всяком случае, на большинстве современных устройств в большинстве современных браузеров, возможно отображение глаголических текстов в их исконном виде.

На смену глаголице в старославянских памятниках приходит кириллица. Версии её происхождения также разнятся: некоторые предполагают постепенное формирование [Гранстрем Е. Э., с. 303], иные – изобретение Константином-Кириллом, как и в случае с глаголицей [Карский Е. Ф., 1979, с. 160]. Так или иначе, непосредственным предком кириллической письменности, по всей видимости, является греческий алфавит [Там же, с. 161].

Кириллических алфавитов существенно больше, чем глаголических. В данной работе рассматривается старославянская кириллица, в которой фиксируется 44 буквы [Вайан А., 1952, с. 29]. Как и в случае с глаголицей, большая их часть обозначает конкретные фонемы старославянского языка. Для глаголицы, как и для кириллицы, характерны йотированные графемы (ср. *ю* в *огню* ‘огню’ [TITUS]). Некоторые фонемы представляются одновременно лигатурой и отдельной графемой (ср. *вашему – вашемѹ* ‘вашему’ [TITUS]). Ряд графем обозначает исконно неславянские звуки или звукосочетания, зачастую используемые исключительно в числовом значении (ср. ~*ѯ*~ ‘60’). Варианты графемы *иже*, *i* и *и*, так же, как и в глаголических памятниках, встречаются в одном и том же слове (ср. *іного – иного* ‘иного’). Звук [ы] в кириллических текстах может обозначаться как сочетанием графем *ъи*, так и графемой *ы* (*тъи – ты* ‘ты’ [TITUS]).

Старославянские кириллические тексты характеризуются более поздним временем появления, чем глаголические, и они не столь многочисленны: кириллица станет активно использоваться после вымирания старославянского. Одни из наиболее крупных старославянских кириллических текстов – Супрасльская рукопись и Саввина книга. Помимо них, сохранились ещё отрывки кириллической надписи на развалинах церкви в Преславе, а также ряд памятников малого объёма, таких как Македонский кириллический листок.

Кириллица также поддерживается стандартом Unicode [Cyrillic Unicode], несмотря на некоторые нюансы [Kempgen S. S., 2008]. При составлении корпуса, во всяком случае, его веб-версии, в данный момент выглядит возможным представление кириллических текстов с использованием системы письменности, при помощи которой они были созданы.

Однако в данный момент большая часть доступных в электронном формате текстов с распознанными символами представлена не в кириллической или глаголической системах письменности. Это тексты, доступные в [CCMH] и [TITUS]; в них каждый кириллический и глаголический символ закодирован через определённый символ или последовательность символов таблицы ASCII [MacKenzie C. E., 1979]. Они наследуют все особенности, характерные для исходной системы письменности.

На латинице представлено сравнительно небольшое количество текстов, однако при этом сами тексты либо играют особенную роль в изучении старославянской письменности, либо выделяются своими размерами. Так, в формат ASCII переведены Киевские листки, самый древний из текстов старославянского канона, Супрасльская рукопись, несколько евангелий, а именно Зографское, Мариинское и Ассеманиево. Саввина книга также представлена в электронном виде с использованием латиницы. Помимо этого, следует отметить ряд текстов, принадлежность к старославянскому канону которых в той или иной степени спорна, в силу лингвистических особенностей или отсутствия полноценного анализа из-за сравнительно позднего обнаружения: Зографское евангелие – Б, жития Константина и Мефодия, Пражские фрагменты.

Формат ASCII универсален и может быть использован для представления текста на любой платформе. Однако при этом этот формат не является изначальным форматом представления текстов, отображаемых с его помощью в существующих электронных коллекциях. С учётом же развития технологий с тех пор, как эти тексты были переведены в электронную форму, преимущества универсальности ASCII нивелируются повсеместным распространением систем, поддерживающих Unicode-кодировки на глаголице и латинице. При составлении нового корпуса тексты, представленные в формате ASCII, будут нуждаться в перекодировке.

**2.3.** Частеречный состав старославянского языка.

Понятию «часть речи» в рамках общего языкознания не дано единственного определения [Алпатов В.М., 1986, с. 36]. Строго говоря, наиболее общим определением будет указание на части речи в целом как достаточно маленькая совокупность *P* (в пределах десяти, при особенно подробной классификации – двадцати) множеств *s1…sn*, такая, что любой токен *t* множества слов *T* языка *L* принадлежит одному и только одному такому классу:

Деление токенов на эти классы при этом кажется абсолютно необходимым для исследования и фундаментального описания любого языка, по каким бы критериям оно проведено ни было [Алпатов В. М., 1986, с. 36]. В работе используются части речи, рекомендуемые для языков мира в [Zeman et al., 2020]. По ним уже было размечено Мариинское евангелие в рамках [TOROT].

В данной работе термин «токен» используется в первую очередь потому, что, во-первых, многим исследователям не менее тяжёлым, чем определение понятия «часть речи» представляется определение понятия «слово», а, во-вторых, не всякая единица языка, релевантная для задачи частеречной разметки, однозначно будет признана словом или его составляющей. Так, в силу разной степени сохранности текстов возможно выделить категорию фрагментарных токенов, в корпусе представленных последовательностями графем, разделённых знаками равенства *=*. Им нельзя присвоить однозначную частеречную принадлежность, но для удовлетворения условий, указанных выше, им присваивается собственный тэг. Также следует выделить в отдельные категории знаки пунктуации и цифры. При составлении корпуса цифры, в силу особенностей их графического выделения (ср. *~ИВ~* ‘12’ [TITUS]), не включались в категорию числительных, как это рекомендуется в работе [Zeman et al., 2020], а составляли отдельную категорию. Непосредственно в категорию чисел были отнесены количественные числительные (*дъва* ‘два’ [TOROT]). Порядковые числительные, как и в [Zeman et al., 2020], были приравнены к прилагательным. Наконец, существует категория для обозначения нестарославянских слов, отдельными синтагмами встречающихся в старославянских текстах, таких как *савахтани* ‘савахфани’ [TOROT].

Критерии разделения слов на части речи могут быть семантическими, морфологическими, синтаксическими и интуитивными [Алпатов В.М., 1986, с. 36 – 43]. В чистом виде ни один из них в классификациях, как правило, не представлен [Там же, с. 36].

Наиболее важным и релевантным исследователям представляется проведение границ между знаменательными частями речи, обладающими лексической самостоятельностью [Алпатов В. М., 1986, с. 36]. При этом решением исследователей из этого множества зачастую исключаются междометия, слова, используемые для эксплицитного маркирования экспрессивной реакции [Zeman et al., 2020]. Примером междометия в старославянском языке может быть *ей* ‘эй!’ [TOROT].

Служебным частям речи уделяется тем меньше внимания, чем меньше объём доступных на языке текстов. Так, в старославянском языке отдельное пристальное внимание исследователи уделяют разве что союзам, и это внимание уделяется исключительно их синтаксическим функциям, но не подробной характеристике [Вайан А., 1952, с. 397]. Союзы, соединяющие определёнными отношениями слова или структуры из нескольких слов, могут быть сочинительными (ср. *i* ‘и’ [TOROT]) и подчинительными (ср. *да* ‘пусть’ [TOROT]), и в рамках ряда подходов выступать не единым, а двумя разными классами [Zeman et al., 2020].

Для определения служебных частей речи в корпусе [TOROT] используется система тэгов проекта Universal Dependencies [Zeman et al., 2020]. К ним относятся в первую очередь предлоги, которые «формируют с именной группой единую структуру, предназначенную для выражения грамматического и семантического отношения к другой единице внутри клаузы» [Там же]. Примером предлога может выступать *въ* ‘в’ [TOROT]. Помимо предлогов, в отдельную категорию выделяются детерминативы. Они используются для эксплицитного выражения определённости (ср. *вьсь* ‘весь’ [TOROT]). На границе между служебными и знаменательными частями речи находятся вспомогательные глаголы, практически потерявшие лексическое значение и выполняющие примарно грамматическую функцию, которую не может взять на себя в силу особенностей словоизменения смысловой глагол [Zeman et al., 2020]. К числу вспомогательных глаголов в старославянском в первую очередь следует отнести *быти* ‘быть’ [TOROT].

Основным противопоставлением выступает оппозиция глагола и остальных знаменательных частей речи. Под глаголом в данном случае подразумевается любое слово, которое «обозначает действие и событие, выступает как минимальный предикат в клаузе и управляет числом и формами других составляющих клаузы» [Zeman et al., 2020], ср. *оставити* ‘оставить’. Причастия в старославянском также традиционно причисляются к этому классу, к примеру, *текъ* ‘тёкший’[TOROT].

Глагол часто сопровождается наречием, указывающим на «время, место, направление, способ» [Zeman et al., 2020]. В старославянском примером наречия может служить *ѕѣло* ‘очень’ [TOROT].

Напрямую противоположено глаголу имя существительное, обозначающее «лицо, место, вещь, животное или абстрактное понятие» [Zeman et al., 2020]. Одним из самых частотных существительных старославянского языка является *богъ* ‘бог’ [TOROT]. В отдельную категорию по различным причинам, от особенностей синтаксического поведения до удобства автоматической обработки при составлении словаря, выносятся имена собственные, такие как *иосифъ* ‘Иосиф’ [TOROT].

Любые существительные, как собственные, так и нарицательные, могут быть модифицированы прилагательными, составляющими отдельную часть речи в старославянском языке, ср. *сѫботьнъ* ‘субботний’ [TOROT].

Наконец, заменять существительные или именные группы могут местоимения, обширно представленные в старославянском языке. Синтаксически они выполняют схожие функции, при этом всегда являясь только референтом, ср. *его* ‘его’ [TOROT].

Таким образом, в рамках корпуса старославянского языка релевантным оказывается деление всего массива токенов на следующие группы: NOUN (имя существительное), PROPN (имя собственное), VERB (глагол), ADJ (прилагательное), ADV (наречие), PRON (местоимение), AUX (вспомогательный глагол), DET (детерминатив), ADP (предлог), SCONJ (подчинительный союз), CCONJ (сочинительный союз), INTJ (междометие), NUM (числительное), X (нестарославянское слово), DIGIT (цифра), PUNCT (знак пунктуации), FRAG (фрагмент).

**2.4.** Выводы.

Таким образом, была представлена краткая теоретическая характеристика старославянского языка.

Указано его положение на генеалогическом древе славянских языков, подчёркнута важность изучения в силу данной специфики. Охарактеризованы периоды развития старославянского языка, а также его основные лингвистические особенности, важные для классификации текстов внутри старославянского канона.

Старославянские тексты изначально записывались глаголицей, затем – кириллицей, и значительно позднее, уже после вымирания языка, часть из них была переведена в подмножество латиницы, путём замены исходных символов на ASCII. При составлении корпуса значительное время будет уделено восстановлению исходной системы письменности для этих текстов, возможному благодаря Unicode.

Определена классификация частей речи. Она отличается от классификаций, приведённых в классических работах, однако она наиболее удобна для автоматической обработки корпуса, уже применена в другом проекте и более полно представляет служебные части речи, а также типы токенов, не являющиеся словами старославянского языка. Этот набор тэгов будет использован при частеречной разметке в дальнейшем.

## **Глава 3. Старославянский язык как объект составления корпуса.**

* 1. Определение текстового состава корпуса старославянского языка.

В силу упоминавшегося выше отсутствия среди исследователей консенсуса по вопросу того, какие именно тексты следует включать в корпус старославянского языка и по каким критериям их следует включать, любое решение подобного характера неизбежно будет носить субъективный характер. Помимо этого, оно будет базироваться на наборе критериев, в том числе таких, которые не являются лингвистическими в строгом понимании этого слова, но скорее историческими.

Последнее замечание относится в первую очередь к тексту времени второй родины [Kamphuis J., 2020, с. 5], Киевским листкам, также известным, как Киевские фрагменты, или Киевский миссал. И именно тексту этого времени, а не написанному в рамках деятельности на территории второй родины. Как упоминалось в теоретической характеристике старославянского языка, в Киевских листках сочетаются черты южнославянских и западнославянских диалектов в области тех ключевых рефлексов, которые непосредственно служат для определения принадлежности текста к старославянскому языку или же изводу церковнославянского языка [Kamphuis J., 2020, с. 5]. Тем не менее, Киевские листки включаются в корпус текстов старославянского языка большей частью исследователей, занимающихся анализом его явлений [Kamphuis J., 2020, с. 11] [СС, 1994, с. 18]. Это делает необходимым включение Киевских листков в состав корпуса старославянского языка практически на основании критерия исторической договорённости исследователей. Критерия, который, таким образом, к данному моменту оказывается не просто решающим, но перевешивающим все остальные при составлении корпуса старославянского языка.

В дальнейшем, когда корпус будет представлен в виде электронной базы данных, станет возможным его более точный лингвистический анализ на гомо/гетерогенность, после чего окажется возможным либо сформировать подкорпуса в соответствии с этими языковыми особенностями, либо переопределить само понятие старославянского языка, основываясь на новых данных. Однако на данный момент определяющим критерием всё же избирается критерий исторического консенсуса.

Тем не менее этот критерий, даже с учётом его очевидной экстралингвистической сущности, не должен быть полностью произвольным, сводящимся к признанию выбора каждого учёного, занимавшегося старославянским языком. Ни одна работа, даже полностью методологически компетентная, признанная классической (можно даже написать: особенно такая работа), не может по отдельности служить абсолютным определителем того, какой текст следует включать в корпус, а какой – нет. Поэтому первичное решение следует принимать на основании совокупности фундаментальных работ, признанных научным сообществом, существующих электронных коллекций текстов и корпусов старославянского языка, а также современных обобщающих работ по старославянскому языку. В дальнейшем, в ходе дискуссии и корпусного анализа, круг текстов будет возможно изменить, если в этом появится необходимость.

При составлении выборки используются следующие работы.

1. Классические исследования старославянского языка – работы, написанные на всём протяжении истории изучения старославянского языка, наиболее подробно её обобщающие. Это прежде всего написанные в середине двадцатого века исследования [Селищев А. М., 1951] [Вайан А., 1952] [Мейе А., 1934].
2. К категории современных фундаментальных исследований старославянского языка мы прежде всего относим [Kamphuis J., 2020], которая уделяет наибольшее внимание характеристике текстов старославянского и выделению их ключевых особенностей с точки зрения ряда критериев, как исторических, так и непосредственно лингвистических, в том числе, рефлексов \**tj/\*dj* [Kamphuis J., 2020, с. 15].
3. Особого внимания заслуживает словарь старославянского языка, представляющий собой обобщение изучения последнего в период до конца второй половины 20 века и включающий в себя перечисление и подробное описание памятников, на основе которых он составлялся [СС].
4. Наконец, следует упомянуть электронные коллекции старославянских текстов, а именно кирилло-мефодиевский хельсинкский корпус [CCMH], построенную во многом на его основе часть корпуса Нового Завета на разных языках [TOROT] и коллекцию Франкфуртского университета [TITUS].

В совокупности эти работы не дают максимально возможно полной картины, однако их наличие необходимо, и, в силу фундаментальности каждой, может претендовать на достаточность. Возможность включения в данную выборку иных исследований представляется предметом дальнейшей дискуссии.

Все работы, попавшие в выборку, считаются равными по своему весу для определения того, стоит ли включать конкретный текст в корпус старославянского языка. Обратный выбор открыл бы возможность для дальнейшей субъективизации критерия, чего, с учётом приведённых выше соображений, допускать не следует.

Тексты в соответствии с критерием будут поделены на три категории.

В первую категорию войдут те тексты, которые присутствуют во всех четырёх категориях. Консенсус в их отношении существует на протяжении по меньшей мере семидесяти лет, а составители корпусов считают необходимым включить в корпус именно их. Помещение их в создаваемый корпус, таким образом, становится достижением необходимого минимума полноты и репрезентативности.

Во вторую категорию войдут те тексты, которые присутствуют в работах хотя бы трёх категорий из оговорённых выше. Они также будут включены в корпус естественным образом, поскольку в их отношении был достигнут достаточно широкий и долговременный консенсус. Однако их включение последует за включением текстов первой группы.

В третью категорию войдут тексты, присутствующие в работах двух или одной категорий из оговорённых выше. Они могут быть включены в корпус старославянского языка вслед за текстами второй категории на основании лингвистического анализа (поиска уникальных для старославянского языка особенностей, в первую очередь, рефлекса *\*tj/\*dj* [Kamphuis J., 2020, с. 15]).

Распределение по категориям представлено в таблице 1.

Таблица 1. Частотность включения текстов в канон старославянского языка исследователями.

|  |  |
| --- | --- |
| Количество включений | Тексты |
| 4 | 1. Зографское евангелие [Селищев А. М., 1951, с. 69] [Вайан А., 1952, с. 20] [Мейе А., 1934, с. 8] [СС, 1994, с. 13] [Kamphuis J., 2020, с. 9] [CCMH] [TITUS] 2. Ассеманиево евангелие [Селищев А. М., 1951, с. 70] [Вайан А., 1952, с. 20] [Мейе А., 1934, с. 8] [СС, 1994, с. 15] [Kamphuis J., 2020, с. 10] [CCMH] [TITUS] 3. Мариинское евангелие [Селищев А. М., 1951, с. 71] [Вайан А., 1952, с. 20] [Мейе А., 1934, с. 8] [Kamphuis J., 2020, с. 9] [СС, 1994, с. 14] [CCMH] [TOROT] [TITUS] 4. Саввина книга [Селищев А. М., 1951, с. 74] [Вайан А., 1952, с. 20] [Мейе А., 1934, с. 8] [Kamphuis J., 2020, с. 10] [СС, 1994, с. 15] [CCMH] [TITUS] 5. Супрасльская рукопись [Селищев А. М., 1951, с. 74] [Вайан А., 1952, с. 21] [Мейе А., 1934, с. 8] [СС, 1994, с. 22] [Kamphuis J., 2020, с. 11] [CCMH] [TITUS] |

*Продолжение таблицы 1. Частотность включения текстов в канон старославянского языка исследователями.*

|  |  |
| --- | --- |
| Количество включений | Тексты |
| 4 | 1. Киевские листки [Селищев А. М., 1951, с. 68] [Вайан А., 1952, с. 22] [Мейе А., 1934, с. 8] [СС, 1994, с. 17] [Kamphuis J., 2020, с. 11] [TITUS] |
| 3 | 1. Клоцов сборник [Селищев А. М., 1951, с. 72] [Вайан А., 1952, с. 21] [Мейе А., 1934, с. 8] [СС, 1994, с. 21] [Kamphuis J., 2020, с. 12] 2. Синайская псалтырь [Селищев А. М., 1951, с. 72] [Вайан А., 1952, с. 21] [Мейе А., 1934, с. 8] [Kamphuis J., 2020, с. 10] [СС, 1994, с. 17] 3. Синайский требник [Селищев А. М., 1951, с. 72] [Вайан А., 1952, с. 21] [Мейе А., 1934, с. 8] [СС, 1994, с. 17] [Kamphuis J., 2020, с. 11] 4. Охридские листки [Селищев А. М., 1951, с. 73] [Вайан А., 1952, с. 20] [Kamphuis J., 2020, с. 10] [СС, 1994, с. 16] 5. Хиландарские листки [Селищев А. М., 1951, с. 77] [СС, 1994, с. 24] [Kamphuis J., 2020, с. 12] 6. Листки Ундольского [Селищев А. М., 1951, с. 77] [Вайан А., 1952, с. 20] [Kamphuis J., 2020, с. 10] [СС, 1994, с. 16] 7. Зографские листки [Селищев А. М., 1951, с. 77] [СС, 1994, с. 25] [Kamphuis J., 2020, с. 12] |

*Продолжение таблицы 1.* *Частотность включения текстов в канон старославянского языка исследователями.*

|  |  |
| --- | --- |
| Количество включений | Тексты |
| 3 | 1. Рыльские глаголические листки (включая Македонский глаголический листок и Лист Григоровича) [Селищев А. М., 1951, с. 73] [Вайан А. , 1952, с. 20] [СС, 1994, с. 24] [Kamphuis J., 2020, с. 12] |
| 1-2 | 1. Надпись на надгробном камне, сделанная в 993 году по распоряжению Самуила, царя юго-западной Болгарии и Македонии [Селищев А. М., 1951, с. 74] [Вайан А., 1952, с. 20] [Kamphuis J., 2020, с. 12] 2. Отрывки кириллической надписи на развалинах церкви в Преславе [Селищев А. М., 1951, с. 74] [Kamphuis J., 2020, с. 12] 3. Македонский листок [Селищев А. М., 1951, с. 77] [Kamphuis J., 2020, с. 12] 4. Надписи в Райградском сборнике [Селищев А. М., 1951, с. 78] 5. Остромирово евангелие [Вайан А., 1952, с. 21] [Мейе А., 1934, с. 8] 6. Никольское евангелие [Вайан А., 1952, с. 21] 7. Ватиканский палимпсест [Kamphuis J., 2020, с. 10] 8. Боянское евангелие-апракос [Kamphuis J., 2020, с. 10] [СС, 1994, с. 16] 9. Синайский фрагмент [Kamphuis J., 2020, с. 10] 10. Енинский апостол [Kamphuis J., 2020, с. 10] [СС, 1994, с. 17] 11. Псалтырь Дмитрия [Kamphuis J., 2020, с. 11] |

*Окончание таблицы 1.* *Частотность включения текстов в канон старославянского языка исследователями.*

|  |  |
| --- | --- |
| Количество включений | Тексты |
| 1-2 | 1. Синайский миссал [Kamphuis J., 2020, с. 11] 2. Санкт-Петербургский октоих [Kamphuis J., 2020, с. 11] 3. Зографский палимпсест [СС, 1994, с. 16] 4. Зографское евангелие – Б [CCMH] [TITUS] 5. Житие Константина [CCMH] [TITUS] 6. Житие Мефодия [CCMH] [TITUS] 7. Пражские фрагменты [TITUS] |

В работе [Селищев А. М., 1951] упоминается также ряд текстов с чертами иных славянских языков, однако их автор рассматривает скорее как вспомогательный материал, нежели как материал непосредственно исследования. Исключением выступают только Киевские листки, которые демонстрируют, по мнению автора, архаичность достаточную, чтобы причислить их к старославянским памятникам [Селищев А. М., 1951, с. 79].

Работа [Вайан А., 1952] настаивает на включении целого ряда текстов, носящих черты иных славянских языков, в группу старославянских. В частности, Остромирова и Никольского евангелий, а также Киевских листков. Объяснение, приводимое автором, носит скорее экстралингвистический характер, акцент делается на «древности и особой важности» [Вайан А., 1952, с. 20 – 21]. В отношении Киевских листков напрямую делается вывод об их принадлежности к моравскому церковнославянскому языку, но они тем не менее включаются в ряд старославянских памятников [Вайан А., 1952, с. 22].

Исследование [Мейе А., 1934] включает Супральскую рукопись и Киевские листки в состав канона старославянского языка, однако при этом делается ряд ремарок. Во-первых, утверждается, что влияние моравских диалектов на Киевские листки весьма значительно, однако второстепенный характер этого памятника позволяет не учитывать его языковые особенности при составлении общей характеристики старославянского языка [Мейе А., 1934, с. 9]. Во-вторых, что Супрасльская рукопись лишена ряда важных архаичных черт, характерных для «языка первых переводчиков», однако не указывается на то, какие именно важные архаичные черты в ней отсутствуют [Мейе А., 1934, с. 9]. Несмотря на это, данные особенности этих памятников в работе не упоминаются как препятствие для их включения в текстовый канон старославянского языка.

В исследовании [Kamphuis J., 2020] также упоминаются Новгородские фрагменты и Слуцкая псалтирь [Kamphuis J., 2020, с. 14], используемые автором в качестве дополнительного материала, однако не являющиеся частью общего старославянского канона. Также автор указывает на то, что Киевские листки могут быть выделены в отдельную традицию в силу объективных лингвистических обстоятельств. Однако они всё же включаются в состав старославянских памятников и им [Kamphuis J., 2020, с. 15].

В источниках одной или двух групп встречаются следующие тексты: надпись на надгробном камне, сделанная в 993 году по распоряжению Самуила, царя юго-западной Болгарии и Македонии, отрывки кириллической надписи на развалинах церкви в Преславе, Македонский (кириллический) листок, надписи в Райградском сборнике, Остромирово евангелие, Никольское евангелие, Ватиканский палимпсест, Боянское евангелие-апракос, Синайский фрагмент, Енинский апостол, Псалтырь Дмитрия, Синайский миссал, Санкт-Петербургский октоих, Зографский палимпсест, Зографское евангелие – Б, Житие Мефодия, Житие Константина, Пражские фрагменты.

Никольское евангелие, как уже упоминалось, было включено автором в собранную им коллекцию в силу экстралингвистических причин [Вайан А., 1952, с. 20 – 21]. Использовать его при проектировании корпуса в условиях отсутствия широкого консенсуса кажется не рациональным. При этом широким консенсусом Остромирово евангелие признано памятником восточнославянского извода церковнославянского языка, его включение будет нарушением цельности корпуса. Аналогично её нарушением будет включение и Пражских фрагментов в коллекции [TITUS]. Они принадлежат чешскому изводу церковнославянского языка, на что указывается непосредственно в источнике [TITUS].

Санкт-Петербургский октоих – документ, известный преимущественно по восточнославянским и южнославянским спискам, а следовательно, подвергшийся значительному влиянию соответствующих языков. По всей видимости, тексты, представленные в нём, являются скорее текстами восточнославянского или какого-либо из южнославянских изводов церковнославянского языка [Крашенинникова О. А., 2006, с. 6]. Октоих, таким образом, лингвистически не является памятником старославянского языка, и, таким образом, не будет помещён в корпус.

Включение Псалтыри Димитрия [Tarnanidis I., 1988] и Синайского миссала [Миклас Х. и др.] в состав корпуса также вызывает сомнения. Их текст не опубликован полностью до сих пор, существуют лишь фрагментарные примеры языка памятников. Следовательно, любой вывод о том, каков именно этот язык, сделать в данный момент времени будет затруднительно. До публикации полного текста включение этих двух текстов в корпус следует отложить.

В случае остальных памятников требуется провести дополнительное исследование, поскольку не существует ни экстралингвистических, ни исторических причин, ни широкого консенсуса для принятия этих документов в корпус старославянского языка, или их исключения из него. В данном случае становится необходимым лингвистический их анализ по критериям, представленным в [Kamphuis J., 2020].

Таблица 2. Лингвистический анализ надписи Самуила [Успенский Ф. И., 1899, с. 3] по критериям [Kamphuis J., 2020].

|  |  |  |
| --- | --- | --- |
| Исходное сочетание | Рефлекс в тексте | Примеры проявления рефлекса в тексте |
| \**tj*/\**dj* | – | – |
| \**óRC* | *раС* | *рабъ* ‘раб’ |
| \**tl*/\**dl* | – | – |

В представленных в таблице 2 данных очевидна нехватка рефлекса, по которому определяется принадлежность к старославянскому языку, и сам текст в целом очень небольшой. По единственному зафиксированному рефлексу \**óRC* > *раС* можно утверждать только то, что это либо южнославянский язык, либо чешский или словацкий, однако не более того. Поэтому включение надписи Самуила в корпус старославянского языка на данный момент вызывает сомнения.

В случае с отрывками кириллической надписи на развалинах церкви в Преславе, изданными в [Медынцева А., Попконстантинов К., 1985], ситуация ещё более затруднительная. Никаких специфически старославянских черт, даже разделённых с другими языками, в ней обнаружено не было. Её нельзя исключить из корпуса на основании критериев [Kamphuis J., 2020], однако нельзя включить по той же причине: невозможно с точностью верифицировать, каким именно языком пользовался автор.

Таблица 3. Лингвистический анализ Македонского кириллического листка [Ильинский Г. А., 1906, с. 5] по критериям [Kamphuis J., 2020].

|  |  |  |
| --- | --- | --- |
| Исходное сочетание | Рефлекс в тексте | Примеры проявления рефлекса в тексте |
| \**tj*/\**dj* | *шт/жд* | *аште* ‘если’, *обрѧштеть* ‘обрящет’  *нѹжди* ‘нужде’, *ижде* ‘иже’ |
| \**óRC* | *раC* | *разѹмѣѥть* ‘понимает’, *разѹмь* ‘разум’, *рабь* ‘раб’ |
| \**tl*/\**dl* | *л* | *молѫ* ‘молю’ |

Язык Македонского кириллического листка, таким образом, является в полной мере старославянским. В первую очередь, сочетания \**tj* и \**dj* последовательно отражаются как *шт* и *жд* на письме, а именно это является главной причиной отнесения текста к старославянскому языку. Подтверждается гипотеза и по двум дополнительным критериям: переход \**órC > раC*, и упрощение группы согласных \**tl*/\**dl* (в данном случае переход \**dl* > *л*). Македонский кириллический листок, таким образом, следует поместить в подготавливаемый корпус старославянского языка.

Таблица 4. Лингвистический анализ надписей в Райградском сборнике [Соболевский А.И., 1910, с. 154 – 161] по критериям [Kamphuis J., 2020].

|  |  |  |
| --- | --- | --- |
| Исходное сочетание | Рефлекс в тексте | Примеры проявления рефлекса в тексте |
| \**tj*/\**dj* | *шт/ж* | *сѫщ(и)* ‘сущий’, *носѧщиiа* ‘носящая’  *рожьсто* ‘рождество’ |
| \**óRC* | *раC* | *раздрушьно* ‘разрушительно’ |
| \**tl*/\**dl* | – | – |

В тексте сравнительно мало рефлексов \**tj*, и всего лишь один рефлекс \**dj*, что в значительной степени затрудняет точность отнесения к определённому языку. Косвенных признаков лингвистической принадлежности текста также практически нет, и на основании их можно заключить, что он либо южнославянский, либо чешский, либо словацкий. Однако существующие описания указывают на то, что большая часть надписей в Райградском сборнике была сделана восточнославянскими писцами в период после старославянского языка. На это, по мнению авторов, указывают, к примеру, вставное *д* в *раздрушьно* ‘разрушительно’, а также написание через йотированный ук имени *кюрилъ* ‘Кирилл’ [Соболевский А. И., 1910, с. 157]. Именно эти косвенные особенности, в совокупности с фиксируемым единственным рефлексом \**dj* > *ж*, а не *жд*, указывают на восточнославянскую природу текста, который, таким образом, не должен быть включён в корпус старославянского языка.

Таблица 6. Лингвистический анализ Синайского фрагмента [Glibetić N., 2015, с. 16 – 19] по критериям [Kamphuis J., 2020].

|  |  |  |
| --- | --- | --- |
| Исходное сочетание | Рефлекс в тексте | Примеры проявления рефлекса в тексте |
| \**tj*/\**dj* | *шт/жд* | *поштѧдi* ‘пощади’, *[свѣ]штѫ* ‘свечой’  *(д)[а]ждь* ‘дай’ |
| \**óRC* | *раC* | *рабъ* ‘рабов’ |
| \**tl*/\**dl* | *–* | *–* |

Текст Синайского фрагмента сравнительно мал и сохранился недостаточно хорошо, в результате чего анализ по критериям [Kamphuis J., 2020] в значительной степени затрудняется. В таблице 6 представлен полный список лексем из текста, которые для этого анализа могут быть использованы. Для одного из дополнительных критериев, рефлекса \**tl*/\**dl*, материала найдено не было. Для второго – единственная лексема. Для основного критерия, рефлекса *\*tj*/\**dj*, три лексемы, но две из них наполовину восстановлены, хотя сами рефлексы, *шт* и *жд*, видны достаточно отчётливо. Те слова, в которых *шт* и *жд* были восстановлены исследователями, в выборку не вошли. Однако по имеющимся признакам текст можно отнести к памятникам старославянского языка.

Таблица 7. Лингвистический анализ Енинского апостола [Мирчев К. и Кодов Х., 1965] по критериям [Kamphuis J., 2020].

|  |  |  |
| --- | --- | --- |
| Исходное сочетание | Рефлекс в тексте | Примеры проявления рефлекса в тексте |
| \**tj*/\**dj* | *шт/жд* | *немощнѹ* ‘немощному’, *могѫщомѹ* ‘могущему’, *нощъ* ‘ночь’  *прѣжде* ‘прежде’, *туждего* ‘чужого’, *wдеждѫ* ‘одежду’ |
| \**óRC* | *раС* | *разумъ* ‘разум’, *раба* ‘раба’, *работѫ* ‘работу’ |
| \**tl*/\**dl* | *л* | *молитвы* ‘молитвы’ |

Текст Енинского апостола лингвистически, по всей видимости, один из самых поздних в выборке. На это указывает прежде всего одноеровое письмо, означающее, что процесс падения редуцированных в языке переводчика или переписчика либо активно проходил, находясь в завершающей фазе, либо завершился к моменту написания. На весь текст не зафиксировано ни одного употребления графемы *ь*, и при этом обнаружено достаточно большое количество слов, в которых редуцированный переднего ряда этимологически восстанавливается. К ним относится, например, *нечъстивии* ‘нечестивые’, *полъsѫ* ‘пользу’, *нощъ* ‘ночь’ и *дънъ* ‘день’ (оба редуцированных этимологически переднего ряда).

Помимо этого, Енинский апостол обладает рядом графических особенностей.

Среди этих особенностей – отказ от употребления графем, обозначающих йотированные носовые гласные. Особенно заметно это становится, когда рассматриваются повторяющиеся юсы большие в окончаниях имён существительных, исторически разделённых йотом. Отражала ли эта графическая особенность реальные фонетические изменения, утверждать представляется затруднительным, однако текст Енинского апостола очень характерно благодаря ей выделяется. В качестве примеров надлежит привести следующие слова: *вѣчнѻѫ* ‘вечную’, *съгрѣшаѫщеи* ‘грешащие’, *законнѫѫ* ‘законную’.

Помимо этого, в тексте несколько раз встречаются глаголические буквы, ср *щ* в *[отъв]рѣщи* на листе 1б издания [Мирчев К. и Кодов Х., 1965]. Это ещё одно указание на сравнительно позднюю природу текста, представляющего собой, по всей видимости, кириллический список глаголического оригинала, сделанный человеком, владевшим обоими алфавитами и при этом не разграничивавшим их употребление. Однако на непосредственно языковые особенности данная черта не указывает.

Однако все особенности, характерные для Енинского апостола, тем не менее, не характеризуют его, как текст какого-либо позднего извода церковнославянского языка. Напротив, лингвистические особенности Енинского апостола указывают на его явную принадлежность старославянскому. Прежде всего, это рефлексы \**tj* и \**dj*, *шт* и *жд* соответственно. Дополнительные признаки, рефлексы \**óRC* и \**tl*/\**dl*, также фиксируют принадлежность текста южнославянской группе (исключая отнесённость поочерёдно к восточной и западнославянской группам) языков. На основании представленных критериев, таким образом, Енинский апостол является текстом старославянского языка и будет включён в корпус.

Следующие тексты, Зографское евангелие – Б и Зографский палимпсест во многом совпадают, в силу того, что Зографское евангелие – Б написано поверх Зографского палимпсеста кириллицей. Тем не менее это два разных текста. Зографское евангелие – Б значительно больше по объёму (это заметно в том числе по разнообразию рефлексов, которые оказалось возможным проанализировать); помимо этого, в совпадающих отрывках пропущены разные фрагменты. По этой причине Зографское евангелие – Б и Зографский палимпсест следует рассматривать и анализировать отдельно.

Таблица 8. Лингвистический анализ Зографского палимпсеста [Добрев, 1971, с. 160 – 163] по критериям [Kamphuis J., 2020].

|  |  |  |
| --- | --- | --- |
| Исходное сочетание | Рефлекс в тексте | Примеры проявления рефлекса в тексте |
| \**tj*/\**dj* | *шт* | *сѣдалишта* ‘то, на чём сидит X’, *продаѭштихъ* ‘продающих’, *аще* ‘если’ |
| \**óRC* | *раC* | *разбоиникомъ* ‘разбойникам’, *рабъ* ‘раб’ |
| \**tl*/\**dl* | *л* | *молитвѣ* ‘молитвы’ |

Затруднением при анализе Зографского палимпсеста выступает полное отсутствие слов, на примере которых можно было бы проверить, как отразилось сочетание \**dj*. Однако высокая частотность рефлекса *шт* < \**tj*, а также дополнительные признаки, рефлексы сочетаний \**óRC* и \**tl*/\**dl*, в целом показывают, что по определяющему критерию рефлекса сочетаний дентальных согласных с йотом Зографский палимпсест с языковой точки зрения является текстом старославянского языка, а следовательно, его включение в корпус окажет положительное влияние на полноту последнего.

Таблица 9. Лингвистический анализ Зографского евангелия – Б [CCMH] по критериям [Kamphuis J., 2020].

|  |  |  |
| --- | --- | --- |
| Исходное сочетание | Рефлекс в тексте | Примеры проявления рефлекса в тексте |
| \**tj*/\**dj* | *шт/жд* | *хощетъ* ‘хочет’, *отвѣщавъ* ‘отвечал’, *обрѧщеши* ‘найдя’  *тѹждiхъ* ‘чужих’, *прѣжде* ‘прежде’, *даждъ* ‘дай’ |
| \**óRC* | *раC* | *раздрѣшите* ‘разрешите’, *рабе* ‘рабы’ |
| \**tl*/\**dl* | *л* | *молѣше* ‘молился’, *помолитъ сѧ* ‘помолиться’, *молитвѣ* ‘молитвы’ |

Набор рефлексов, представленных в Зографском евангелии – Б, достаточно обширен, и позволяет сделать вывод о том, что текст действительно принадлежит старославянскому канону. Особенно важно, что представлены рефлексы и \**tj* > *шт*, и \**dj* > *жд*, позволяющие сделать однозначный вывод о языке памятника. Зографское евангелие – Б на основании этого будет включено в корпус старославянского языка.

Таблица 10. Лингвистический анализ жития Мефодия [CCMH] по критериям [Kamphuis J., 2020].

|  |  |  |
| --- | --- | --- |
| Исходное сочетание | Рефлекс в тексте | Примеры проявления рефлекса в тексте |
| \**tj*/\**dj* | *шт*, *ч/ ж* | *аште* ‘если’, *ношти* ‘ночи’*, помошть* ‘помощь’  *чюжеи ‘*чужой*’*  *ражаѥть* ‘рожает’, *прѣже* ‘прежде’, *межю* ‘между’*, дажь* ‘дай’, *жажи* ‘жажды’ |
| \**óRC* | *раC, лаС, алС* | *разѹмѣти (*и иные слова с приставкой *раз*- ‘раз-’ <\**orz*) ‘понимать’, *равнъ* ‘равный’, *раби* ‘рабы’  *лакомьства* ‘лакомства’  *алъкани* ‘алчности’ |
| \**tl*/\**dl* | *л* | *молитвы* ‘молитвы’ (и иные формы слова *молитва* ‘молитва’) |

Можно заметить, что по дополнительным признакам текст принадлежит к староцерковнославянскому языку, или, по меньшей мере, к группе церковнославянских языков. Групп согласных *тл, дл < \*tl*, \**dl* не зафиксировано (в случае активного влияния чешской традиции вместо форм слова *молитва* использовались бы формы слова *modlitwa*). Рефлекс *RoС* < \* *óRC* в тексте также не присутствует, что говорит по меньшей мере о доминировании южнославянского субстрата в тексте над потенциальным восточнославянским влиянием: текст не является полным переводом на древневосточнославянский. На это же указывает и встречающийся рефлекс \**tj* > *шт*, однако другие данные вызывают сомнения в том, какой именно звук обозначает графема *шта* в тексте.

В частности, восточнославянские рефлексы \**dj* > *ж* и *\*tj* > *ч*, ср. *чюжеи* < *чужъ* ‘чужой’ и ст.-слав. *штоуждь* ‘чужой’. Рефлекса \**dj* > \**жд* в тексте не встречается вообще, как следствие, житие Мефодия в известном списке не является текстом старославянского языка. Однако теперь требуется установить, на каком именно языке оно написано: дополнительные признаки указывают на достаточно сильный южнославянский субстрат, это и не древневосточнославянский текст в полной мере. Следует предположить, что перед нами восточный извод церковнославянского языка. На это указывает несколько обстоятельств.

В первую очередь следует упомянуть такие лексемы, как *бештиньны* ‘бесчинствующий’и *бештисльнъ* ‘бесчисленный’[CCMH]. Исходно в них на месте буквы *шта* восстанавливается сочетание [с’ч’]. Возможность слияния этих двух фонем означает, что в языке переписчика (на то, что мы имеем дело с поздним списком, указывается непосредственно в исходном корпусе [CCMH]) уже существовал звук [ɕː], который не мог существовать в старославянском языке, однако существовал в ряде славянских языков, формировавшихся в тот же период. Чтобы установить язык в точности, одного этого показателя будет недостаточно. Однако в совокупности с рефлексом \**dj* > *ж* это наводит на мысль о том, что язык текста находился под активным влиянием древневосточнославянского.

Вторым важным фактором в определении языка памятника становится редкое использование юса малого сразу перед назальным согласным [н], ср.: *Костѧнтинъмь цѣсарьмь* ‘Константином царём’[CCMH]*, Селоунѧнина* [CCMH]. В других контекстах мы видим его использование на месте сочетания «гласный + согласный», трансформировавшиеся в носовой гласный переднего ряда [ę], ср. *отьць великаго цѣсарѧ Костѧтина* [CCMH]. Ещё одна группа представляет собой слова, в которых этимологически восстанавливается звук [‘а] там, где писец поставил юс малый, ср.: *нарицѧемаго, нарицѧти* [CCMH]. Наконец, присутствует в тексте и словоформа *iазыка* [CCMH]. Здесь явление носит обратный характер. В месте, где этимологически восстанавливается йотированный носовой гласный переднего ряда, ср. польское *język* ‘язык’, писец использует йотированный аз. Исходя из этого, можно утверждать, что в языке переписчика произошло падение носовых гласных, и рефлексом падения носового гласного переднего ряда стал звук [‘а]. Это служит ещё одним доказательством того, что переписчик может быть носителем древневосточнославянского языка.

Последним важным явлением, которое необходимо охарактеризовать, становится полногласный рефлекс –*TороT*– <\**TarT*, появляющийся в словоформе *королѧ* [CCMH]. Это явление носит исключительно восточнославянский характер, а следовательно, и список создавался на территории проживания восточных славян, значительно позже периода существования старославянского языка.

Графема *шта*, исходя из сделанных выводов, скорее всего, обозначает не звук [ш’т’]. Она представляет собой средство фиксации на письме мягкого шипящего [ɕː], восходящего или к этимологически восстанавливаемой исконно старославянской аффрикате [ш’т’] в таких словах, как *помошти* ‘помощи’ (ср. рус. *помощи* ‘помощи’), или к сочетанию [с’ч’] в словах любого происхождения, ср. *бештисльнъ* ‘бесчисленный’.

Таким образом, можно утверждать, что гипотеза верна, и в списке представлен текст восточнославянского извода церковнославянского языка, пусть и восходящий к старославянскому источнику. Это подтверждается и данными других исследователей, чья работа была посвящена текстологической, исторической и лингвистической характеристике памятника [Grivec F., Tomšić F*.,* 1960, с. 83].

С учётом всех приведённых выше данных в отношении жития Мефодия можно с точностью утверждать, что данный памятник, являясь памятником восточнославянского извода церковнославянского языка, понизит степень репрезентативности и гомогенности корпуса старославянского языка, а следовательно, не должен быть включён в данный корпус.

Таблица 11. Лингвистический анализ жития Константина [CCMH] по критериям [Kamphuis J., 2020].

|  |  |  |
| --- | --- | --- |
| Исходное сочетание | Рефлекс в тексте | Примеры проявления рефлекса в тексте |
| \**tj*/\**dj* | *шт(щ)/жд* | *хощеть* 'хочет’, *аще* ‘если’, *искѹшающе* ‘искушая’, *помощию* ‘помощью’, *wтвѣщаше* ‘отвечая’, *пасѹщiи* ‘пасущие’  *туждь* ‘чужой’, *даждь* ‘дай’, *ѹгаждаѥ* ‘угождает’, *вижду* ‘вижу’, *между* ‘между’, *граждань* ‘горожан’ |
| \**óRC* | *раС, лаС* | *разумь* ‘разум’, *разлучи* ‘разлучил’, *рабь* ‘раб’, *разоренѹ* ‘разорённому’, *различiи* ‘различия’  *лакомiа* ‘лакомого’ |
| \**tl/*\**dl* | *л* | *молiаше* ‘моля’, *молитвахь* ‘молитвах’, *молити* ‘молить’ |

Житие Константина представляет собой памятник со значительно меньшим количеством восточнославянских черт, чем житие Мефодия. Рефлексы сочетаний \**tj* и \**dj* характерны для восточной части южнославянского ареала, принадлежность языка памятника к ней косвенно подтверждается рефлексами \**óRC* и \**tl/*\**dl*. В тексте не зафиксировано случаев, когда этимологически восстанавливаемое сочетание *сч* передаётся графемой *щ*, а также лексем, в которых были бы обнаружены неюжнославянские рефлексы прежних сочетаний с плавными. Данный текст может быть назван старославянским по критериям [Kamphuis J., 2020].

Однако в его языке была обнаружена одна особенность, которая вызывает сомнения в правильности гипотезы. Это графемы, которые обнаружены на местах этимологически восстанавливаемых носовых гласных. Там, где ожидается написание юса большого, появляется *у* или *ѹ*, йотированного юса большого – *ю*, юса малого и йотированного юса малого – *е* или *ѥ* (непоследовательно, ср. в тексте *ѥзыкь* ‘язык, народ’ – *езыкь* ‘язык, народ’). Единственный появляющийся юс обнаружен в контексте *въпрошьшем' жѧ нѣкоимь* ‘спросившем же некоего’, где, по всей видимости, обозначает не носовой гласный, а его рефлекс, встречающийся по всему тексту.

По всей видимости, текст является очень поздним списком старославянского оригинала [Grivec F., Tomšić F*.,* 1960, с. 83], сделанным в южнославянском ареале (на последнее указывает рефлекс носового переднего ряда). Его характерные языковые особенности не позволяют включить его в корпус.

Анализ Ватиканского палимпсеста и Боянского евангелия в данный момент провести возможным не представляется, поскольку тексты для исследователей недоступны из-за эпидемиологической ситуации в мире в момент написания работы.

* 1. Автоматическая предобработка некоторых текстов старославянского языка.

Большая часть текстов, находящихся в данный момент в электронном доступе, доступна исключительно в транслитерации латиницей в кодировке ASCII. Исходно эти тексты могли быть написаны как кириллицей, так и глаголицей. При их помещении в корпус, следовательно, крайне желательно представить их в исходном формате.

Однако в этот момент появляется новое ограничение. Проведение автоматической разметки в данный момент достаточно сильно зависит от того, совпадает ли система письменности массива, на котором обучается тэггер, и массива, на котором он используется. Единственный размеченный набор данных старославянского языка на данный момент – текст Мариинского евангелия, доступный в рамках Universal Dependencies [Zeman и др., 2020]. Там он записан кириллицей. Следовательно, первой системой письменности, в которую в рамках корпуса будет переводиться любой текст, поступающий на вход, станет именно старославянская кириллица.

При расшифровке текстов, не переведённых в машиночитаемую форму и доступных лишь в формате сканированных страниц (как Енинский апостол), кириллица выбирается как алфавит с самого начала. Однако тексты, доступные в коллекции TITUS, требуют по меньшей мере перевода из кодировки ASCII в кодировку Unicode. При этом некоторые проблемы, такие как восстановление этимологических редуцированных в оригинальных публикациях, не разрешимы методами автоматической обработки в целом. Однако непосредственно перевод из кодировки в ASCII в кодировку Unicode, а также токенизация результатов, выглядит вполне возможным.

Исходный текст, к примеру, одной из строк в Киевских листках, будет выглядеть приблизительно следующим образом: *\*nebes$sk~yE tv'oE sJl^y* ‘небесные твои силы’ [TITUS]. Для помещения его в корпус требуется провести определённое количество преобразований, чтобы, в конечном итоге, получить строку, похожую на *Небесъскыѥ твоѥ сiлы*.

После того, как текст получен из места исходного размещения, в первую очередь проводятся перестановки вспомогательных символов, указывающих на диакритические знаки, и астериска, указывающего на заглавность следующей графемы. К примеру, последовательность символов *\*`* изменяется на *`\**. Эта операция выполняется для более эффективного приведения букв к их прописному варианту.

Затем проводится смена кодировки буквенных символов в соответствии с соответствующей транслитерационной таблицей [CCMH]. При наличии перед графемой астериска используется её заглавный вариант, ср. *t&gda* ‘тогда’ > *тъгда*, но \**t&gda > Тъгда*.

В некоторых случаях возникает необходимость провести программу по тексту два раза, поскольку транслитерационные таблицы даже в одном источнике, как правило, не унифицированы, и одна и та же последовательность символов ASCII может трактоваться по-разному. Так, при транслитерации некоторых текстов *U* обозначало ук, а других – ижицу. Следовательно, *Upokriti* ‘лицемеры’ в Мариинском евангелии [CCMH] должно транслитерироваться как *ѵпокрити*, а *iskUSeni* ‘искушённые’ в Супрасльской рукописи [CCMH] – как *искушени*. Были также составлены правила, позволяющие идентифицировать разные обозначения ижицы, *U* и *Y* (используемая только в предположениях составителей) в Мариинском евангелии. Ижица распознаётся и в *отъпо[ѵ]сті* ‘отпусти’, и в *сѵдонъскѫ* ‘сайдасской/сидонской’ [CCMH].

На следующем этапе предобработки синтагмы, разорванные между двумя строками, соединяются в верхней строке: *тво-\nего* ‘тво-\nего’ [TITUS], где \n – знак переноса строки, превращается в *твоего*.

После этого проводится непосредственно токенизация. Особое внимание при ней уделяется анализу фрагментарных токенов по правилам, приведённым в работе [Афанасьев И. А., 2020, с. 18]. Была предпринята попытка использовать токенизатор пакета NLTK [Bird и др., 2009], однако он с данной задачей не справился, в силу отсутствия подходящей модели или системы правил, предназначенной для обработки фрагментов.

Затем проводится дополнительное разделение текста на фрагменты по знакам пунктуации.

На финальном этапе предобработки текст помещается в базу данных, в данный момент представленную коллекцией .json-файлов на компьютере пользователя. Каждый .json-файл представляет собой отдельный документ в корпусе, хранящийся в файле как объект класса *Document*. Каждый *Document* обладает идентификатором (поле *documentID*),названием (поле *fileName*), метаданными (поле *documentMetaData*), опционально – путём к файлу с нераспознанным текстом .pdf в системе и путём к этому же файлу в Google Docs (поля *filePath* и *googleDocPath* соответственно). Внутри *Document* хранится список объектов класса *Text*. Этот список может состоять из одного объекта, если документ сравнительно небольшой, или из многих объектов, к примеру, отдельных листков. *Text* также обладает идентификатором, названием и метаданными (поля *textID, textName* и *textMetaData*). Этот объект разбивается на список объектов типа *Clause*, устроенного схожим образом и дробящимся на список объектов типа *Realization*, представляющих собой отдельные токены. Они обладают набором полей разметки *realizationFields*, двумя полями представления *lexemeOne* и *lexemeTwo*, необходимыми для корректного отображения в корпусе, а также состоят из списка графем *Grapheme*, каждая из которых является атомарной единицей базы данных.

Поля *lexemeOne* и *lexemeTwo* отличаются различным отображением диакритических знаков в исходном тексте. Поле *lexemeOne* сохраняет их нетронутыми, позволяя исследователю в дальнейшем присвоить графемам, идущим за маркерами диакритических знаков или перед ними (в зависимости от указаний в таблице транслитерации), соответствующие значения полей разметки. Поле *lexemeTwo* стирает их, представляя в корпусе текст, визуально более близкий к представленному в оригинальном документе. Также в поле *lexemeTwo* знак *!* заменяется на *~*, если токен не является цифрой. Определение последнего проходит по соответствующему регулярному выражению. Оно может быть записано следующим образом: токен начинается с восклицательного знака, за которым может следовать или не следовать буква, обозначающая сотни, за которой может следовать или не следовать буква, обозначающая десятки, за которой может следовать или не следовать буква, обозначающая единицы, за которой может следовать или не следовать буква, обозначающая цифру 10, за которой обязательно следует конец токена. Если токен удовлетворяет этому правилу, тогда (и только тогда) он окаймляется знаками ~, как этого требуют таблицы транслитерации. При этом важно заметить, что не всегда цифра в исходном тексте маркируется титлом, что значит, что перед ней при перекодировке в ASCII не ставится знак *!*. Это влечёт за собой то обстоятельство, что некоторые числа, к примеру, *В* ‘2’ [TITUS] не распознаются в дальнейшем как числа, что создаёт необходимость ручного корректирования результатов дальнейшей разметки.

После того, как этот последний этап обработки завершается, текст уже доступен в корпусе, однако существуют ещё две задачи, которые предстоит решить исследователю

* 1. Частеречная разметка документов в корпусе старославянского языка.

Для обучения модели был использован текст Мариинского евангелия, размеченный в рамках проекта [TOROT] в формате Universal Dependencies [Zeman и др., 2020]. В этом наборе данных насчитывается порядка 50 тыс. токенов, разделённых на обучающее, валидирующее и тестовое подмножества. С учётом имплементации нашей модели частеречной разметки [HMM] обучающее и валидирующее подмножества использовались как единое, оценка эффективности процесса обучения проводилась на тестовом подмножестве.

Было проведено четыре серии экспериментов. В первой был определён бейзлайн: результаты тестирования обученной на болгарском языке, одном из ближайших современных родственников старославянского, модели TreeTagger [Helmut S., 1994]. Вторая серия включала несколько экспериментов на *n-*граммных моделях. Третья представляет собой попытки гибридизации скрытой марковской и *n-*граммной моделей. Использовался набор данных UD. Четвёртый эксперимент проводился на материале Киевских листков и заключался в сравнении эффективности использования скрытой марковской модели и наиболее эффективного из гибридов.

Для определения бейзлайна использовались параметры модели, обученной не на старославянском, поскольку предварительно обученных параметров TreeTagger для старославянского не существует, а исходный код данной модели не находится в открытом доступе. Таким образом, бейзлайн определяется результатами модели, приблизительно напоминающими гибридную архитектуру, представленную ниже, обученной на материале языка, отдалённо похожего на старославянский, связанного с ним генетически, но не типологически. Была предпринята попытка адаптировать модели, обученные на русском и словацком языках, однако их точность достигала всего лишь 26% и 1% соответственно, в силу критических различий в наборах тэгов.

В таблице 1 представлены соответствия наборов тэгов UPOS[UD UPOS tagset] и болгарских параметров TreeTagger [Simov K и др., 2004].

Таблица 12. Сопоставление тэгов TreeTagger-bg (TT) и UD.

|  |  |
| --- | --- |
| UD | TT |
| ADJ | A, Mo, Md, My, H |
| ADV | D |
| INTJ | I |
| NOUN | Nc |
| PROPN | Np |
| VERB | Vn, Vp |
| ADP | R |
| AUX | Vx, Vy, Vi |
| CCONJ | Cc, Cr, Cp |
| DET | Ps |
| NUM | Mc |
| PART | T |
| PRON | Pp, Pd, Pr, Pc, Pi, Pf, Pn |
| SCONJ | Cs |
| X | (если не размечено) |

В таблице 13 представлены результаты модели, используемой в качестве бэйзлайна, по каждому из тэгов. Эти результаты – минимум, который должна побить любая модель, работающая на старославянском материале.

Таблица 13. Подсчёт точности частеречной разметки моделью TreeTagger-bg (TT) на старославянском наборе данных из UD, общей и уточнённой по каждому тэгу. Результаты здесь и далее округлены до второй цифры после запятой.

|  |  |
| --- | --- |
| PoS | TT |
| VERB | 78.65 |
| AUX | 1.85 |
| ADV | 2.41 |
| NOUN | 30.49 |
| PRON | 4.34 |
| CCONJ | 36.92 |
| ADP | 25.97 |
| ADJ | 25.67 |
| INTJ | 8.47 |
| SCONJ | 2.17 |
| DET | 3.73 |
| PROPN | - |
| NUM | - |
| X | - |
| **Total** | 31.54 |

В качестве базовой модели использовался счётчик n-грамм. Эта модель находила самые частотные n-граммы для каждой части речи в тренировочном наборе данных. В случае со старославянским языком, использовались две самых частотных *n-*граммы. Во время фазы предсказания, модель осуществляет их поиск в каждом из токенов, если же их нет, то присваивает тэг VERB, самый частотный в обучающем и тестовом наборах данных (8356 и 2281 токенов соответственно, 10637 в общей совокупности. Результаты для различных *n* (2, 3, 4) представлены в таблице 14. 3-граммная модель продемонстрировала наибольшую эффективность, хотя она едва превосходила бэйзлайн, и нуждалась в дальнейших улучшениях.

Таблица 14. Подсчёт точности частеречной разметки для моделей TreeTagger-bg, 2-граммной, 3-граммной и 4-граммной. Лучшие результаты в сравнении, здесь и далее, выделены **полужирным**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PoS | TT | 2-grams | 3-grams | 4-grams |
| VERB | 78.65 | 43.93 | 96.02 | **98.88** |
| AUX | 1.85 | - | 77.85 | **96.04** |
| ADV | 2.41 | **70.16** | 67.08 | 67.08 |
| NOUN | **30.49** | 16.4 | - | - |
| PRON | 4.34 | 66.09 | 83.92 | **87.83** |
| CCONJ | 36.92 | **45.91** | 20.66 | 20.66 |
| ADP | 25.97 | **53.35** | 49.48 | 49.59 |
| ADJ | 25.67 | 51.64 | **98.8** | 63.49 |
| INTJ | 8.47 | **19.38** | 18.12 | 18.12 |
| SCONJ | 2.17 | 31.66 | **51.55** | 48.97 |
| DET | 3.73 | 18.68 | **43.9** | - |
| PROPN | - | **5.26** | - | - |
| NUM | - | - | - | - |
| X | - | 19.66 | **20.82** | 20.26 |
| **Total** | 31.54 | 30.25 | **32.19** | 31.43 |

N-граммы преимущественно были первыми *n* символами токенов конкретной части речи. Средняя дистанция первого и последнего символов токена до первого символа *n-*граммы дана в таблице 15.

Таблица 15. Среднее распределение *n*-грамм для некоторых частей речи.

|  |  |  |
| --- | --- | --- |
| PoS | Расстояние от начала | Расстояние от конца |
| VERB | 0 | 8.1 |
| ADJ | 0.29 | 6.91 |
| X | 0 | 6 |
| Среднее | 0.07 | 7.79 |

Были рассмотрены различные возможности улучшения 3-граммной модели, поскольку для некоторых частей речи, таких как *ADJ*, она давала достаточно высокие результаты. Например, была имплементирована основанная на правилах система, которая нормализует сокращения, находящиеся под титлом в оригинальных текстах. Однако простое удаление титла показалось более полезным, поскольку сокращения, такие как *iс* (представляющее самое частотное имя собственное в корпусе, *icоусъ* ‘Иисус’), распознаются моделью с большей вероятностью. В начало и конец слова добавлялся специальный символ *#*, что увеличило эффективность в общей сложности почти на семь процентов. Также были предприняты дополнительные попытки усилить модель с использованием длины токена как критерия определения части речи, обозначения диграфа *оу* [у] и повторяющихся символов как одного символа в рамках *n*-граммы, и использования взвешивания частотности *n*-грамм по метрике TF-IDF. TF-IDF взвешивание было подсчитано при помощи следующей формулы, где 0 ≤ *k* < l – выбранный пользователем коэффициент нормализации, *C* – показатель встречаемости данной *n*-граммы в рамках определённой части речи, *N* – максимальное число *n*-грамм в массиве токенов набора данных, *L* – общее число токенов в наборе данных, а *Q* – номер токенов, содержащих данную *n*-грамму:

Создаётся впечатление, что критерий длины токена оказался слишком неоднозначным, диграфы и повторяющиеся символы встречались слишком редко, чтобы изменить *status quo* для двух наиболее частотных *n*-грамм, а взвешивание по метрике TF-IDF никак не повлияло на оценку частотности столь постоянно встречающихся *n*-грамм.

Результаты эксперимента представлены в таблице 16.

Таблица 16. Подсчёт точности частеречной разметки 3-граммной моделью (3-gram), усиленной декапитализацией (3 + D), декапитализацией и обозначением границ токенов (3 + DB), последними двумя и взвешиванием по метрике TF-IDF (3 + DBT), либо подсчётом длины токенов каждой части речи (3 + DBL), либо подсчётом повторяющихся символов и диграфов как единых символов (3 + DBR).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PoS | 3-gram | 3 + D | 3 + DB | 3 + DBT | 3 + DBL | 3 + DBR |
| VERB | 96.02 | **96.13** | 54.96 | 54.96 | 51.84 | 54.96 |
| AUX | 77.85 | 77.85 | **100** | 96.3 | **100** | **100** |
| ADV | 67.08 | **67.39** | 61.86 | 61.86 | 16.97 | 61.86 |
| NOUN | - | - | **37.99** | **37.99** | 36.79 | **37.99** |
| PRON | 83.92 | 83.92 | **89.22** | **89.22** | - | **89.22** |
| CCONJ | 20.66 | 21.09 | **85.16** | **85.16** | 46.95 | **85.16** |
| ADP | 49.48 | **49.49** | 40.49 | 40.49 | - | 40.49 |
| ADJ | 98.8 | **98.8** | 86.02 | 86.02 | 88.51 | 86.02 |
| INTJ | 18.12 | 18.57 | **20.16** | **20.16** | - | **20.16** |
| SCONJ | 51.55 | 51.55 | 78.13 | 78.13 | **80** | 78.13 |
| DET | **43.9** | **43.9** | 18.75 | 18.75 | 26.55 | 18.75 |
| PROPN | - | - | 93.81 | 93.81 | **94.64** | 93.81 |
| NUM | - | - | 50 | **56.9** | 45 | 50 |
| X | 20.82 | 20.83 | 23.74 | 23.67 | **28.01** | 23.74 |
| **Total** | 32.19 | 32.45 | **39.17** | 39.15 | 38.27 | **39.17** |

После получения таких результатов, оказавшихся лишь ненамного выше бейзлайна, следующая серия экспериментов была проведена с скрытой марковской моделью, усиленной алгоритмом Витерби [Uludoğan G]. Рассматривалось также использование тэггеров, основанных на рекуррентных нейросетях, и базовых регрессионных моделей, однако, они требовали или значительно большего количества данных, или эмбеддингов, предварительно обученных на столь же большом количестве данных. Эти модели также продемонстрировали тенденцию к переобучению. Это критический недостаток, поскольку обучающий набор данных состоит только из одного текста, Мариинского евангелия, а остальные старославянские тексты могут значительно отличаться от него лингвистически. Возможность модели к адаптации на новых данных приобретает особенную важность, и поэтому данные методы не были имплементированы в рамках модели. Результаты тестирования скрытой марковской модели на Мариинском евангелии представлены в таблице 17.

Таблица 17. Подсчёт эффективности частеречной разметки с использованием скрытой марковской модели, усиленной алгоритмом Витерби.

|  |  |
| --- | --- |
| PoS | HMM |
| VERB | 68.79 |
| AUX | 98.77 |
| ADV | 60.31 |
| NOUN | 99.04 |
| PRON | 86.51 |
| CCONJ | 99.65 |
| ADP | 98.7 |
| ADJ | 56.36 |
| INTJ | 91.53 |
| SCONJ | 70 |
| DET | 73.13 |
| PROPN | 87.33 |
| NUM | 92.56 |

*Продолжение таблицы 17.*

|  |  |
| --- | --- |
| X | 20 |
| **Total** | 81.04 |

Как можно заметить, и скрытая марковская модель, и *n*-граммная модель предвзяты в отношении конкретной части речи. Скрытая марковская модель значительно лучше определяет существительные, однако иногда испытывает затруднения с отличением их от других частей речи. Аналогичное утверждение можно сделать и относительно *n*-граммной модели и глагола. *N*-граммная модель может быть использована эффективнее при определении конкретной части речи (X, глагола или прилагательного). Вместе с тем, её применение может негативно сказаться на общем качестве разметки. Это было учтено на новом этапе исследования.

Он включал различные методы гибридизации между скрытой марковской и n-граммной моделями: путь, проложенный TreeTagger [Helmut, 1994]. При этом полученные модели возможно было адаптировать к структуре старославянских данных за счёт создания систем на правилах.

Все гибриды функционировали по одной и той же схеме. После того, как скрытая марковская модель осуществляла предварительную разметку, 3-граммная модель, с модификациями либо без них, дополнительно проверяла токены, присваивая им одну из тех частей речи, которые она определяла с большей эффективностью, чем скрытая марковская модель. Однако первый тест показал, что делегирование проверки наречий 3-граммной модели снижает общую эффективность. Если же 3-граммная модель дополнительно определяла только прилагательные, глаголы и Х, то точность гибрида повышалась, хотя и ценой снижения точности определения существительных. Некоторое увеличение эффективности, достигаемое добавлением декапитализации, сохранялось. Напротив, обозначение границ токенов привело к лёгкому падению эффективности. Почти каждый гибрид увеличил точность определения Х. Это связано с тем фактом, что в тестовом присвоенных тэгов.

Последний эксперимент включал в себя использование регрессионной модели, выбиравшей результат, основываясь на предсказаниях скрытой марковской и n-граммной. Регрессионная модель использовала метод обучения дерева решений, ранее доказавший свою эффективность на различных соревнованиях по машинному обучению [Strobl и др., 2009]. Точность её колебалась, по всей видимости, в связи с имплементацией округления чисел в Python. И даже в лучшем случае они не превышали результатов скрытой марковской модели.

Из всех различных комбинаций наиболее эффективным себя показал гибрид скрытой марковской и 3-грамной моделей, с декапитализацией токенов во время и фазы обучения, и фазы тестирования, а также с приоритизированием присваивания наречий для скрытой марковской модели. Результаты экспериментов представлены в таблице 18.

Таблица 18. Подсчёт эффективности частеречной разметки с использованием скрытой марковской модели, усиленной алгоритмом Витерби (HMM) и 3-граммной моделью (HMM + 3), определяющей прилагательные, глаголы и Х (HMM + 3 – ADV); последней моделью, усиленной декапитализацией (HMM + 3 + D – ADV), обозначением границ токенов (HMM + 3 + DB – ADV); моделью с декапитализацией, усиленной регрессионной моделью (HMM + 3 + D – ADV + ETR).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PoS | HMM | HMM  + 3 | HMM  + 3  – ADV | HMM  + 3 + D  – ADV | HMM  + 3 + DB  – ADV | HMM  + 3 + D + ETR  – ADV |
| VERB | 68.79 | 71.37 | 72.69 | **72.82** | 55.46 | 72.69 |
| AUX | 98.77 | **98.77** | **98.77** | **98.77** | **98.77** | **98.77** |
| ADV | 60.31 | **80.06** | 60.31 | 60.31 | 60.12 | 57.42 |
| NOUN | 99.04 | 94.66 | 98.08 | 98.08 | 96.16 | 98.02 |

*Продолжение таблицы 18.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PRON | **86.51** | 72.79 | **86.51** | **86.51** | 86.13 | 73.32 |
| CCONJ | **99.65** | **99.65** | **99.65** | **99.65** | **99.65** | 6.83 |
| ADP | **98.7** | 98.18 | 98.18 | 98.18 | 82.47 | **98.7** |
| ADJ | 56.36 | 52.94 | **56.9** | **56.9** | 55.08 | 56.36 |
| INTJ | **91.53** | **91.53** | **91.53** | **91.53** | **91.53** | **91.53** |
| SCONJ | **70** | 63.04 | **70** | **70** | **70** | 19.57 |
| DET | **73.13** | **73.13** | **73.13** | **73.13** | 64.93 | 25.37 |
| PROPN | **87.33** | 86.3 | **87.33** | **87.33** | 83.22 | **87.33** |
| NUM | **92.56** | 91.74 | 91.74 | 91.74 | 90.08 | 63.64 |
| X | 20 | **60** | **60** | **60** | **60** | 20 |
| **Total** | 81.04 | 80.6 | 81.79 | **81.82** | 75.85 | 69.62 |

Последняя стадия включает в себя тестирование на одном из текстов, которые планировалось помещать непосредственно в корпус. Это Киевские листки, неотъемлемая часть старославянского канона, обладающая целым рядом, как указывалось выше, специфических лингвистических особенностей. Киевские листки прежде не размечались полностью, хотя некоторые попытки уже достаточно успешно проводились [Eckhoff H. M., Berdicevskis A., 2015]. Для эксперимента текст был взят из коллекции [TITUS] и предобработан по правилам, указанным выше. Для распознавания знаков пунктуации и чисел была имплементирована дополнительная система правил. Итоговое сравнение проводилось между моделями HMM + 3 + D – ADV, продемонстрировавшей лучший результат в предыдущих экспериментах, и HMM. Улучшенная модель показала результат в 50.93% точности, использовавшаяся в качестве бейзлайна – 32.64%. Обе модели продемнострировали большее количество ошибок, чем на тренировочном наборе данных. Примеры размеченных токенов Киевских листков даны в таблице 19.

Таблица 19. Примеры размеченных токенов из текста Киевских листков.

|  |  |  |  |
| --- | --- | --- | --- |
| Токен | HMM | HMM  + 3 + D  – ADV | Ручная разметка |
| *Твоемѹ* ‘твой-DAT/LOC’ | NOUN | ADJ | ADJ |
| *~ИВ~* ‘12’ | NOUN | DIGIT | DIGIT |
| *.* ‘.’ | NOUN | PUNCT | PUNCT |
| *приведетъ* ‘вести-FUT(I)’ | NOUN | VERB | VERB |
| *присно* ‘всегда’ | NOUN | VERB | ADV |
| *приснодѣвѣ* 'Богородица-DAT/LOC’ | NOUN | VERB | NOUN |

Итоговая модель в данный момент находится в открытом доступе [HMM].

* 1. Лемматизация токенов старославянского языка.

Модель лемматизации соединет основанные на правилах и словаре модули с рекуррентной нейронной сетью (RNN), создавая гибридную систему, способную справиться с большей частью задач, с которой она сталкивается на указанных выше наборах данных.

Система может использовать наивный стемминг, дополнительную информацию (частеречный тэг анализируемого слова), созданный в процессе обучения словарь, а также предсказания, основанные на *n-*граммах. Код лемматизатора доступен на GitHub [Lemmatiser].

В процессе работы модели первой используется система на правилах, присваивающая лемму *===*, если токен имеет часть речи FRAG (фрагмент), либо же копирует токен, если он принадлежит частям речи PUNCT (знак пунктуации), или DIGIT (число). Затем используется словарь, в котором леммам сопоставлены токены, соответствовавшие им в тренировочном наборе данных. Если правила не применимы к токену, и он не встречался в тренировочном наборе данных, то нейронная сеть заданной пользователем архитектуры делает предсказание. При занесении текста в корпус предсказание помещается в соответствующий файл, при тестировании модели – оценивается.

В случае с Киевскими листками оценка происходила вручную, в случае с Мариинским евангелием – автоматически.

Автоматическая оценка состоит из двух этапов.

На первом этапе происходит подсчёт точности предсказания для каждого токена. Используются абсолютная точность, а также метрики сходства строк, а именно расстояния Джаро-Винклера [Jaro M. A., 1989] [Winkler W. E., 1990], Левенштейна [Levenshtein V. I., 1966] и Дамерау-Левенштейна [Damerau F. J., 1964]. Метрики сходства строк были введены для улучшения понимания того, насколько именно точна модель: несмотря на то, что абсолютная точность по-прежнему активно используется [Milintsevich S., Sirts K., 2020] [Akhmetov и др., 2020], в связи с простотой её имплементации, средняя ошибка в одну букву может крайне негативно сказаться на получаемых через неё результатах.

Измерение абсолютной точности было имплементировано независимо, меры сходства строк – при помощи существующих, предварительно оптимизированных библиотек Python. Для измерения расстояния Левенштейна предполагалось использовать два альтернативных решения. Однако при исследовании исходного кода одного из них, библиотеки FuzzyWuzzy[FuzzyWuzzy], было обнаружено, что оно не в точности отвечает определению указанной метрики. В результате применялась только одна имплементация каждой из метрик.

Наиболее существенные ошибки системы обнаруживаются на втором этапе при помощи поиска выбросов [Grubbs F. E., 1969] в результатах измерения точности по расстояниям Джаро-Винклера, Левенштейна и Дамерау-Левенштейна. Затем они классифицируются и анализируются вручную.

В качестве бейзлайна использовалось присваивание лемме токена. Например, леммой слова *грѣси* ‘грех-PL-NOM’ становилась *грѣси* ‘грех-PL-NOM’, но не *грѣхъ* ‘грех’.

Был проведён ряд экспериментов, в ходе которых подсчитывалась точность модели с различными параметрами.

Наиболее существенные изменения (увеличение эффективности на 25% по сравнению с бейзлайном) были привнесены в первую очередь моделью B256, с увеличенным размером мини-батча (256), при этом дробящую на *n*-граммы и токены, и соответствующие им леммы в тренировочной фазе, а также генерирующую лемму по *n*-граммам.

Рост при увеличении мини-батча обосновывается увеличением способности модели к генерализации на более крупных объёмах входных данных, и при этом отсутствием чрезмерной генерализации на слишком крупных и разрозненных объёмах.

Если модель по *n-*грамме токена предсказывает *n-*грамму леммы, но не всю лемму целиком, то предсказание оказывается более эффективным в силу большего соответствия двух последовательностей друг другу.

Ещё одним важным изменением стало увеличение *n*-грамм до максимально возможного размера. Это связано с тем, что модель получает значительно меньше идентичных последовательностей на входе и выходе, и значительно больше – различающихся. Модель теряет возможность вычислить конкретные изменения, однако сам тот факт, что последовательности имеют тенденцию значительно сильнее изменяться в конце, становится более очевидным для неё. Это отражено в ряде моделей B256, 5, 6, 8 и Max (использующими 3-граммы, 5-граммы, 6-граммы, 8-граммы и *n­*-граммы, где *n* – максимальная длина токена в наборе данных, соответственно). Модель Max была протестирована против бейзлайна, показав существенно более высокие результаты на трёх метриках из четырёх. То, что бейзлайн продемонстрировал лучший результат по среднему расстоянию Левенштейна, может быть объяснено некоторыми особенностями имплементации измерения последнего, а также тем, что распределение результатов модели-бейзлайна ближе к нормальному, в то время как модель 8 либо совершает точное предсказание, либо выдаёт совершенно несопоставимый с правильным результат.

Результаты экспериментов представлены в таблице 20.

Таблица 20. Результаты подсчёта точности моделей, осуществляющих лемматизацию токенов старославянского языка, на Мариинском евангелии. A – точность, L – расстояние Левенштейна, D-L – расстояние Дамерау-Левенштейна, J-W – расстояние Джаро-Винклера. N – результаты на нормализованных данных измерений сходства строк, R – результаты на необработанных данных. Все результаты округлены до сотых. Лучшие результаты выделены **полужирным.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Metrics | Baseline | B256 | 5 | 6 | 8 | Max |
| A | 29.24% | 53.96% | 58.67% | 63.86% | 78.81% | **85.67%** |
| L (R) | **3.86** | 5.41 | 5.36 | 5.18 | 4.77 | 4.37 |
| L (N) | **3.73** | 5.31 | 5.26 | 5.07 | 4.7 | 4.34 |
| D-L (R) | 1.77 | 3.21 | 2.96 | 2.58 | 1.56 | **0.88** |
| D-L (N) | 1.74 | 3.2 | 2.96 | 2.5 | – | – |
| J-W (R) | 0.77 | 0.67 | 0.72 | 0.74 | 0.82 | **0.9** |
| J-W (N) | 0.68 | 0.67 | 0.72 | 0.74 | – | – |

Выбросы, полученные во время тестирования модели 13, формировали отдельный набор данных. Их важность заключается в том, что они подчёркивают самые значительные недостатки системы, проблемы с архитектурой, показанные разными метриками. Выбросы также могут продемонстрировать ненормально высокое качество модели, что невозможно получить при работе только с данными, полученными при анализе непосредственно метрик.

Анализ выбросов начинается с поиска лемм, при генерации которых модель чаще всего ошибается. Они совпадают для расстояний Дамерау-Левенштейна и Джаро-Винклера: *имѣти* ‘иметь’, *глаголати* ‘говорить’, *клѧти* ‘проклинать’, *приѩти* ‘принять’, *мънасъ* ‘лицо’, *исходити* ‘исходить’, *чловѣчьскъ* ‘человеческий’, *оумыти* ‘умывать’, *рыба* ‘рыба’, *въходити* ‘входить’. Для расстояния Левенштейна они значительно отличаются: *съвѣдѣтельствовати* ‘свидетельствовать’, *пророчьствовати* ‘пророчествовать’, *законооучитель* ‘тот, кто учит Закону’, *отъдесѧтьствовати* ‘наложить десятину’, *сѵрофѵникиссаныни* ‘жительница Сирии родом из Финикии’, *дальманоуфаньскъ* ‘далмануфанский’, *четврьтовластьць* ‘тетрарх’, *благословлѥнъ* ‘благословлен’. Разница объясняется тем, что результаты подсчёта расстояния Левенштейна демонстрируют большую степень нормальности распределённости в целом, поэтому выбросы среди них реже. Самые частотные выбросы в измерениях расстояния Левенштейна между токеном и леммой совпадают с наименее частотными выбросами в измерениях расстояний Дамерау-Левенштейна и Джаро-Винклера.

Самые частотно генерируемые выбросы более схожи. Для каждой из метрик первые два – *и* ‘и’ и *новъ* ‘новый’. В случае с расстоянием Левенштейна зафиксированы ещё три: *абоѥ* (значение неясно), *оудъ* ‘часть тела’, *къ* ‘к’. Выбросы по расстояниям Джаро-Винклера и Дамерау-Левенштейна совпадают: *божии* ‘божий’, *ѡ* ‘о’, *быти* ‘быть’, посълати ‘послать’, *рѫбъ* (значение неясно), *тъ* ‘тот’, *оу* ‘у’, *исоусъ* ‘Иисус’.

Модель как правило совершает ошибку из-за генерации более коротких последовательностей на выходе по сравнению с корректными. Средняя длина корректного токена на выходе меньше существенно больше средней длины выброса (7.14, 7.14, 16 против 4.02, 4. 02, 2.1 для расстояний Джаро-Винклера, Дамерау-Левенштейна и Левенштейна соответственно). Максимальная длина леммы равна 18, максимальная длина выброса – 12.

Для тестирования модели Max на Киевских листках, тексте, значительно отличающемся от других старославянских текстов на *n*-граммном уровне [Kamphuis J., 2020], и потому наиболее показательным для оценки способности модели к генерализации, были добавлены правила для лемматизации фрагментов, знаков пунктуации и цифр. Примеры представлены в таблице 21.

Таблица 21. Примеры токенов, лемматизированных моделью Max с добавлением системы правил.

|  |  |
| --- | --- |
| Токен | Лемма |
| *======абъ* | *===* |
| *:.:* | *:.:* |
| *л'ѣта* ‘год-GEN’ | *лѣто* ‘год’ |
| *НАДЪ* ‘над’ | *гда* (временной аффикс) |

* 1. Выводы.

Таким образом, определён полный состав корпуса старославянского языка и порядок включения в него текстов. На первом этапе будут добавлены Зографское евангелие, Ассеманиево евангелие, Мариинское евангелие, Супрасльская рукопись, Саввина книга и Киевские листки. На втором этапе постепенно будут включаться Клоцов сборник, Синайская псалтырь, Синайский требник, Охридские листки, Хиландарские листки, листки Ундольского, Зографские листки, Рыльские глаголические листки, Македонский кириллический листок, Синайский фрагмент, Енинский апостол, Зографский палимпсест и Зографское евангелие – Б. Существует вероятность включения Ватиканского палимпсеста и Боянского евангелия. Итогом работы должен стать корпус из 19 текстов, с возможным расширением до 21 текста.

Тексты будут постепенно добавляться в корпус после проведения предварительной обработки по выработанным в ходе исследования правилам.

При проведении частеречной разметки и лемматизации на Мариинском евангелии предложенные модели оказываются менее эффективными, чем обученные на множестве языков модели Universal Dependencies, однако демонстрируют лучшие результаты при работе с текстом Киевских листков. В ходе этих экспериментов в очередной раз возникла необходиость поставить вопрос о лингвистической гетерогенности старославянского канона. Полученная модель кроссидиомной частеречной разметки может показать, насколько именно эти различия велики.

Анализ наиболее значительных ошибок модели показал необходимость формирования наборов данных, состоящих из более длинных токенов и лемм, для моделей с seq2seq-архитектурой. Один из таких планируется сформировать в ходе дальнейшей работы с корпусом.

## **Глава 4. Программная реализация корпуса старославянского языка.**

На данный момент корпус старославянского языка имеет двухмодульную структуру.

Первый модуль автоматически обрабатывает тексты, выложенные в открытый доступ ранее, преобразуя их в нужный для хранения в корпусе формат. Он состоит из трёх подмодулей, проводящих предобработку и токенизацию, частеречную разметку и лемматизацию соответственно. Каждый из этих трёх подмодулей может функционировать независимо. Модули частеречной разметки и лемматизации при этом требуют наличия текста, преобразованного в нужный для хранения в корпусе формат.

Второй модуль предназначен для ручной обработки текстов, ранее или недоступных в электронной форме, или доступных только в качестве сканированного нераспознанного текста, а также для представления обработанных и представленных в необходимом формате текстов. Второй модуль состоит из подмодулей обработки и представления, каждый из которых выполняет свою работу независимо. Для представления текстов их необходимо хранить в определённом формате.

В данной главе даётся краткая характеристика особенностей реализации данных модулей.

* 1. Модуль предобработки.

Модуль предобработки выполнен в форме приложения Windows Forms, написанного на C#, соединённого с двумя программами на Python.

Приложение на C# используется для создания удобного графического интерфейса, через который пользователь может взаимодействовать с запускаемыми через терминал программами на Python, для проведения предобработки текста, а также для сохранения результатов работы модулей в базу данных. Исходный код модуля предобработки выложен в репозитории на GitHub [Preprocessor].

Стартовое окно приложения (рис. 1) на C# отображает две кнопки. Каждая из них открывает одно из основных окон, в которых проходит работа исследователя.

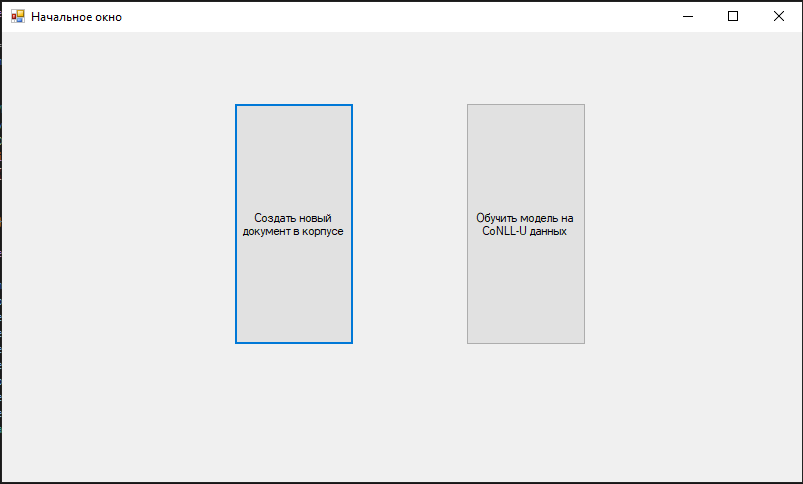


Рис. 1. Стартовое окно модуля предобработки.

Окно, открывающееся по нажатию на кнопку «Обучить модель на CoNLL-U данных», также содержит в себе две кнопки, изображённые на рисунке 2.

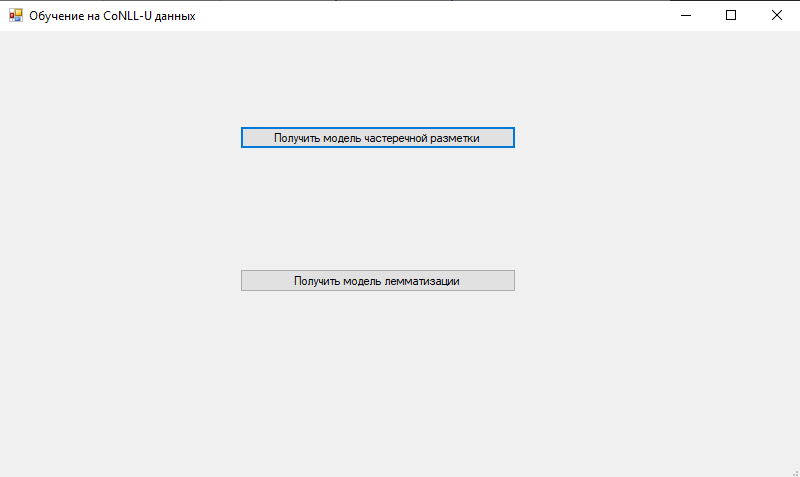


Рис. 2. Окно обучения моделей модуля предобработки.

Первая из них предназначена для связи с подмодулем частеречной разметки для тренировки модели на существующем золотом стандарте, доступном в рамках проекта Universal Dependencies [Zeman и др., 2020].

Подмодуль частеречной разметки располагается в отдельной директории и представляет собой программу на Python. Код подмодуля частеречной разметки состоит из нескольких файлов и доступен в соответствующем репозитории на GitHub [HMM].

Файл *main.py* содержит код, управляющий подмодулем через считывание аргументов командной строки и маршрутизацию к потокам работы. Файл *data\_preparation.py* включает функции для извлечения и работы с тренировочными данными, векторизации *n-*грамм, а также их взвешивания по TF-IDF метрике. В файле *metrics.py* содержатся способы подсчёта точности и отображения матрицы ошибок. Файл *n\_gram.py* включает функционал *n-*граммной модели, *tree\_tag.py* – код с использованием пакета treetaggetwrapper, который осуществляет взаимодействие между кодом Python и программой TreeTagger. В файл *subcategorization.py* внесены изменения по сравнению с оригинальным кодом [Uludoğan G], а именно – добавлены правила для разметки фрагментарных токенов, знаков пунктуации и цифр. Основной код подмодуля содержится в файле *hmm.py*, который включает в себя имплементацию обучения скрытой марковской модели с алгоритмом Витерби, измерения её точности, точности гибридных моделей, а также предсказания и соревновательного предсказания текстов из корпуса.

Подмодуль частеречной разметки может запускаться независимо из терминала командой *python main.py* *--data [fileName]* *[опциональные аргументы]*, где *fileName* – имя файла с расширением *.conllu* (все потоки работы, кроме предсказания) или *.json* (предсказание). К опциональным аргументам относятся следующие:

* *--folder*: основная рабочая директория, в которой размещаются результаты деятельности модели, прежде всего, данные о соревновательном предсказании и подсчёте точности модели при тестировании. По умолчанию используется папка, в которой находится файл *main.py*.
* *--modus*: определяет один из пяти возможных потоков работы: обучение с сохранением объекта модели в файле с расширением .pkl (*training*), тестирование (*accuracy*), соревновательное предсказание (*competitive\_prediction*) – сравнение двух моделей на файле из корпуса, предсказание и сохранение в корпус результатов (*prediction*) и получение векторов слов (*vectorization*). По умолчанию – обучение.
* *--method:* определяет конкретную модель, которая используется в потоке обучения или тестирования: *tt* (TreeTagger)*, hmm* (скрытая марковская модель), *grams* (n-граммная модель), *hmmg* (гибрид последних двух моделей), *hmmc* (регрессионная модель, обучающаяся на результате взаимодействия n-граммной и скрытой марковской моделей). По умолчанию – скрытая марковская модель.
* *--train\_data*: путь к дополнительным тренировочным данным для регрессионной модели.
* *--split*: доля в процентах тренировочных данных по отношению к валидационным во время обучения скрытой марковской модели. По умолчанию аргумент равен 90 процентам.
* *--unknown\_to\_singleton*: отсутствие деления неизвестных токенов на *n-*граммы во время обучения скрытой марковской модели. По умолчанию аргумент равен 0, любой токен делится на *n-*граммы.
* *--printSequences*: вывод результатов обучения в командную строку во время работы скрытой марковской модели. По умолчанию аргумент равен 0, результаты обучения не выводятся в командную строку.
* *--grammage: n* для *n-*граммной модели, количество знаков в последовательностях, на которые разделяется каждый токен. По умолчанию равен 3, так как в ходе тестирования 3-граммная модель показала самые высокие результаты.
* *--register\_change*: определяет, проводится ли декапитализация каждого токена. По умолчанию параметр равен единице: декапитализация, соответственно, проводится.
* *--start\_end\_symbols*: при значении 0 (по умолчанию, рекомендуется при использовании гибридных моделей) в начало и конец слова не добавляется ничего, при значении 1 (рекомендуется при использовании *n*-граммной модели) – специальный символ ‘#’.
* *--weighed*: при обучении *n-*граммной модели частотности могут или взвешиваться с коэффициентом 0 ≤ k < 1, задаваемым аргументом *–tf-idf-coefficient* (по умолчанию равным 0.5) по адаптированной TF-IDF метрике (1), или не взвешиваться (0). При обучении модели, использовавшейся в частеречной разметке корпуса, частотности не взвешивались.
* *--length:* определяет использование (1) (или неиспользование (0)) средней длины токена как дополнительного критерия при определении его части речи. При частеречной разметке корпуса этот параметр не использовался.
* *--double*: с использованием (1) данного параметра повторяющиеся символы и лигатура *оу* воспринимаются программой как единый символ. По умолчанию данный параметр не используется, ему задано соответствующее значение 0.
* *--lang*: параметр для подсчёта точности на существующих весах TreeTagger. Может принимать значения *bg* (болгарский), *ru* (русский) и *sk* (словацкий). Веса, полученные при обучении на болгарском корпусе, используются по умолчанию, в силу их наибольшей эффективности.

При запуске подмодуля частеречной разметки из окна обучения создаются два файла – *hmm.pkl* и *3grams.pkl*. Первый представляет собой веса скрытой марковской модели, второй – *n*-граммной модели. Эти файлы будут использоваться в дальнейшем, при предсказании частей речи для токенов текстов, включаемых в корпус. Также модель может генерировать список векторов (поток векторизации), а также результаты соревновательного предсказания двух моделей на наборе данных в формате *.json* (поток соревновательного предсказания).

Нажатие на вторую кнопку в окне обучения приводит к активации связи с подмодулем лемматизации. Он, как и модуль частеречной разметки, написан на Python. Код подмодуля лематизации также разбит на несколько файлов и доступен в соответствующем репозитории на GitHub [Lemmatiser].

В файле *main.py* так же, как и в случае с подмодулем частеречной разметки, содержится точка входа и дальнейшая маршрутизация. Файл *dataset\_loading.py* включает в себя функции, каждая из которых позволяет загружать отдельный тип набора данных: для обучения, усиления нейросети словарём, или же предсказания. Файл *data\_preprocessing.py* содержит различные способы обработки данных, такие как наивный стемминг, объединение в последовательность слова и его частеречного тэга, разделение слова на *n-*граммы. Помимо этого, в данный файл включена функция, осуществляющая поиск выбросов в результатах подмодуля при подсчёте точности. В файле *metrics.py* содержится имплементация подсчёта расстояния Дамерау-Левенштейна. Непосредственно имплементация модели, потоков работы обучения, подсчёта точности и предсказания находится в файле *model.py*.

Подмодуль лемматизации также запускается из терминала (тестирование проводилось в Windows PowerShell на Windows 10). Приложение активируется после ввода команды *python main.py* *--data [fileName]* *[опциональные аргументы]*, где *fileName* – так же, как и в случае с подмодулем частеречной разметки, имя файла с расширением *.conllu* (потоки работы обучения и подсчёта точности) или *.json* (предсказание). Опциональные аргументы в данном случае используются иные:

* *--folder:* папка, в которую моделью сохраняются файлы, являющиеся результатом её работы, к примеру, веса обученной на тренировочной данных модели. По умолчанию совпадает с папкой, в которой размещается файл *main.py*.
* *--name*: имя модели, задаваемое пользователем для различения результатов функционирования разных архитектур. В процессе обучения в модуле предобработки используется базовое, *seq2seq*.
* *--modus:* предполагаемый пользователем поток работы, исполняемый программой. По умолчанию используется обучение (*training*), также возможны *prediction* (предсказание) и *accuracy* (измерение точности предсказаний работы моделей).
* *--join:* принимает значение 0 (по умолчанию) или 1, во втором случае на вход модели подаётся токен, соединённый с изначально присвоенной ему частью речи.
* *--grams:* определяет *n* для *n-*грамм, на которые делится слово. По умолчанию этого деления не происходит, аргумент передаётся равным 0.
* *--lemma\_split:* при значении 0 на *n-*граммы не делится и предполагаемая лемма, при значении 1 (выбранном по умолчанию после проведения ряда экспериментов) этого не происходит.
* *--stemming:* определяет, проводится ли (1), или же нет (0, по умолчанию) при обучении модели наивный стемминг, позволяющий проводить обучение только на окончаниях словоформ (или на супплетивных словоформах).
* *--forming\_priority:* определяет, как, при делении на по-*n-*граммном предсказании лемм объединяются результаты: присоединением к первой цельной *n-*грамме последнего символа каждой следующей (*forward*, по умолчанию), или же выбором первого символа каждой *n-*граммы вплоть до последней (*back*), включающейся целиком
* *--epochs:* количество эпох, на протяжении которых модель обучается. По умолчанию, чтобы избежать переобучения, равняется 40.
* *--early\_stopping:* при стабилизации результатов, демонстрируемых моделью на обучающей выборке, в течении задаваемого этим аргументом числа эпох, возможно прервать работу модели, чтобы избежать переобучения. По умолчанию эта возможность не используется.
* *--batch:* размер батчей, частей, на которые разделяются обучающие данные. Может варьироваться от 1 до количества токенов, которые подаются на вход модели. Рекомендуется использование базовой конфигурации, 256 единиц в каждом батче.
* *--dim:* количество скрытых измерений модели. По умолчанию равняется 256.
* *--optimizer:* оптимизатор, используемый в модели. Поддерживаются как пользовательские варианты (при условии, что пользователь предоставит соответствующую функцию), так и предоставляемые библиотекой Keras. Базовый оптимизатор – RMSProp, вызываемый значением параметра *rmsprop*.
* *--loss:* функция потери. Так же, как и оптимизатор, может быть имплементирована пользователем самостоятельно, или при помощи Keras. Изначально используется кросс-энтропия, *categorical\_crossentropy*.
* *--activation*: функция активизации. Возможно использование пользовательской, либо уже реализованной в Keras. Функция активации по умолчанию – логистическая функция, обобщённая для многомерного случая, *softmax*.

Когда подмодуль запускается из окна обучения, создаются три файла: *lemmatized\_[name].txt* – словарь зафиксированных пар токен – лемма*, train\_[name].txt* – вспомогательный файл для работы модели, и *[name].h5* – файл с архитектурой модели и обученными весами. При запуске модели для проверки точности из командной строки создаётся файл *errors\_[name].csv*, содержащий информацию об аномально точных и аномально неточных предсказаниях модели на тестовой части выборки.

После подготовки моделей, осуществляющих частеречную разметку и лемматизацию, становится возможным добавление новых текстов в корпус. Модуль предобработки предоставляет один из способов осуществить это через соответствующее окно (рис. 3).

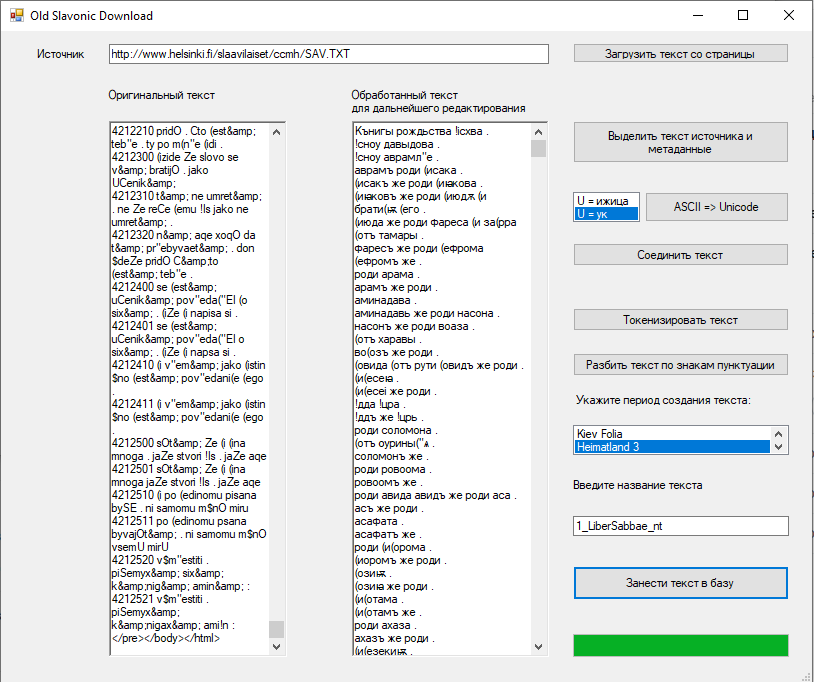


Рис 3. Окно добавления нового текста модуля предобработки.

В верхнее поле вводится ссылка на страницу с текстом (в примере – на Саввину книгу [CCMH]). Текст в необработанном формате загружается в поле «оригинальный текст». Если исходный текст встроен во фрейм (при скачивании копии страницы с web.archive.org), то в поле он вручную заменяется кодом фрейма, скопированным из панели разработчика в браузере.

По нажатии на кнопку «Выделить текст источника и метаданные» текст очищается от HTML-тэгов и обозначений позиций строк в исходном документе. Результаты всех дальнейших действий отображаются в поле «Обработанный текст для дальнейшего редактирования». В случае наличия каких-либо артефактов перекодировки или дополнительной токенизации пользователь имеет возможность вмешаться в работу программы и внести ручные исправления. Ход работы программы отображается на полосе прогресса (на рис. 3 отображено состояние выполненной операции).

Затем производится выбор варианта перекодирования текста из ASCII в Unicode. Из соответствующего поля выбирается, интерпретируется ли U в исходном тексте, как ижица (как в тексте Ассеманиева евангелия), или как ук (как в тексте Саввиной книги). Если символ не встречается в исходном тексте, как в случае с Киевскими листками, выбирается любой из вариантов. Перекодирование осуществляется при помощи регулярных выражений [Thompson, 1968], заменяющих последовательности символов ASCII на символы Unicode.

Текст, полученный в результате, очищается от символов переноса строки путём подъёма второй части перенесённой лексемы на строку выше. Затем проводится дополнительная токенизация, задачей которой является отделение частей фрагментарно распознанного текста, которые могут представлять из себя отдельные токены. После этого текст разбивается на отдельные синтагмы по знакам пунктуации.

Осуществляется базовая метаразметка. В соответствующем поле вводится название текста, а в окне выбора выше – период его создания. Выбор для старославянских текстов осуществляется из четырёх позиций: первой родины (*Heimatland\_1*), второй родины (*Heimatland\_2)*, третьей родины (*Heimatland\_3)*, а также пункта «Киевские листки» (*Kiev\_Folia*) в силу особенности происхождения данного текста. После чего нажатием на кнопку «Занести текст в базу» производится соответствующее действие.

После его завершения пользователю предлагается осуществить частеречную разметку и лемматизацию. При выборе этой опции тексту присваивается тэг *Automatically\_tagged*, и он обрабатывается подмодулями частеречной разметки и лемматизации поочерёдно, при этом пользователю необходимо указать, где располагаются эти подмодули с предварительно обученными ими моделями. Если эта опция не выбрана, тексту присваивается тэг *Not\_tagged*, после чего тот сохраняется в выбранную пользователем директорию. В дальнейшем разметку текста возможно продолжить через отдельное использование подмодулей частеречной разметки и лемматизации.

Когда текст занесён в базу данных (в данный момент представляющую собой простую коллекцию файлов), становится возможным дальнейшая работа с ним из модуля представления.

* 1. Модуль представления.

Концепт этой части системы был представлен в работе [Afanasev, 2020], где вкратце охарактеризовывались основные её части. В ходе создания корпуса старославянского языка было задействовано их ограниченное подмножество, а именно подмодуль создания поля, подмодуль аннотирования текста, подмодуль визуализации текстовых особенностей и подмодуль поиска по корпусу.

Подмодуль создания поля в данный момент находится в стадии доработки, однако его функциональность уже включает в себя, прежде всего, создание полей, снабжённых описанием, присваиваемых всему памятнику, его подразделу, сегменту подраздела, токену, а также графеме, и возможность связать поля между собой, которая будет использована при расширении морфологической разметки. Поля могут иметь заданное пользователем множество значений (как поле периода создания), или произвольное значение, присваиваемое каждой единице базы данных в процессе разметки (как поле леммы). Они могут присваиваться один раз в рамках одного варианта разметки (как поле части речи), либо же несколько раз (в данный момент для корпуса старославянского языка такие поля не используются). Связываться между собой могут только поля одного уровня (например, поле токена с полем токена). Окно подмодуля создания поля представлено на рис. 4.

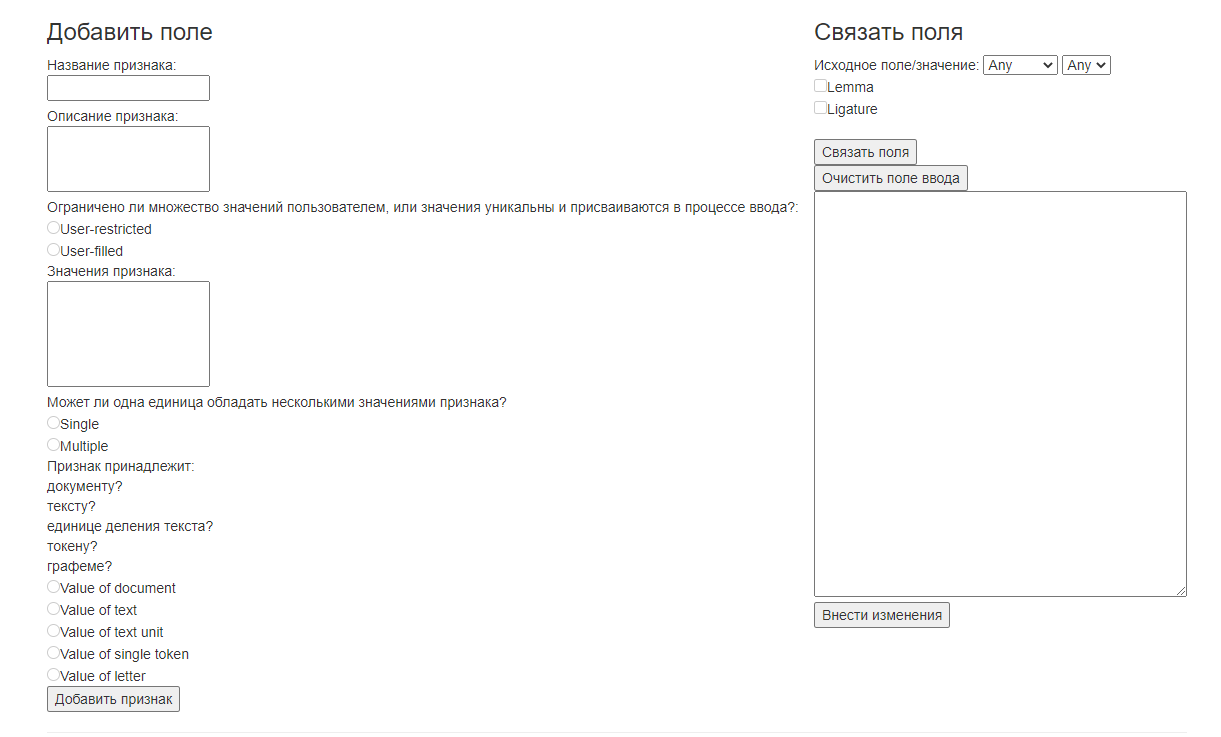


Рис 4. Окно работы с полями модуля представления.

Подмодуль аннотирования текста отображает выбранную пользователем секцию рукописи, разделённую HTML-разметкой на сегменты, которые делятся на токены, которые, в свою очередь, делятся на графемы. Также предоставляется возможность добавить разметку для сегмента, токена и графемы соответственно, причём совпадающие графически токены возможно разметить единовременно, через самый первый из встреченных. Помимо этого, для токенов предусмотрен функционал, позволяющий разметить одну единицу различными способами, на случай, если однозначно установить ту или иную категорию невозможно.

Разметка сегмента осуществляется при нажатии на соответствующую кнопку: токена – при двойном нажатии левой кнопки мыши на него, графемы – при нажатии правой кнопки мыши на ней. В каждом из случаев пользователю предоставляется возможность выбора только тех полей, которые соответствуют уровню аннотируемой единицы. Помимо этого, если выбранное значение выбранной единицы связано с другими полями, ниже предоставляется возможность осуществить её разметку сразу же и по этим полям.

В процессе разметки может потребоваться особое внимание к единицам некоторых классов, либо же к единицам, содержащим в себе некоторую последовательность графем. Для этого в верхней части страницы доступна возможность фильтрации токенов по наличию или отсутствию выбранных пользователем признаков, либо последовательностей символов.

Окно подмодуля аннотирования текста представлено на рис. 5.

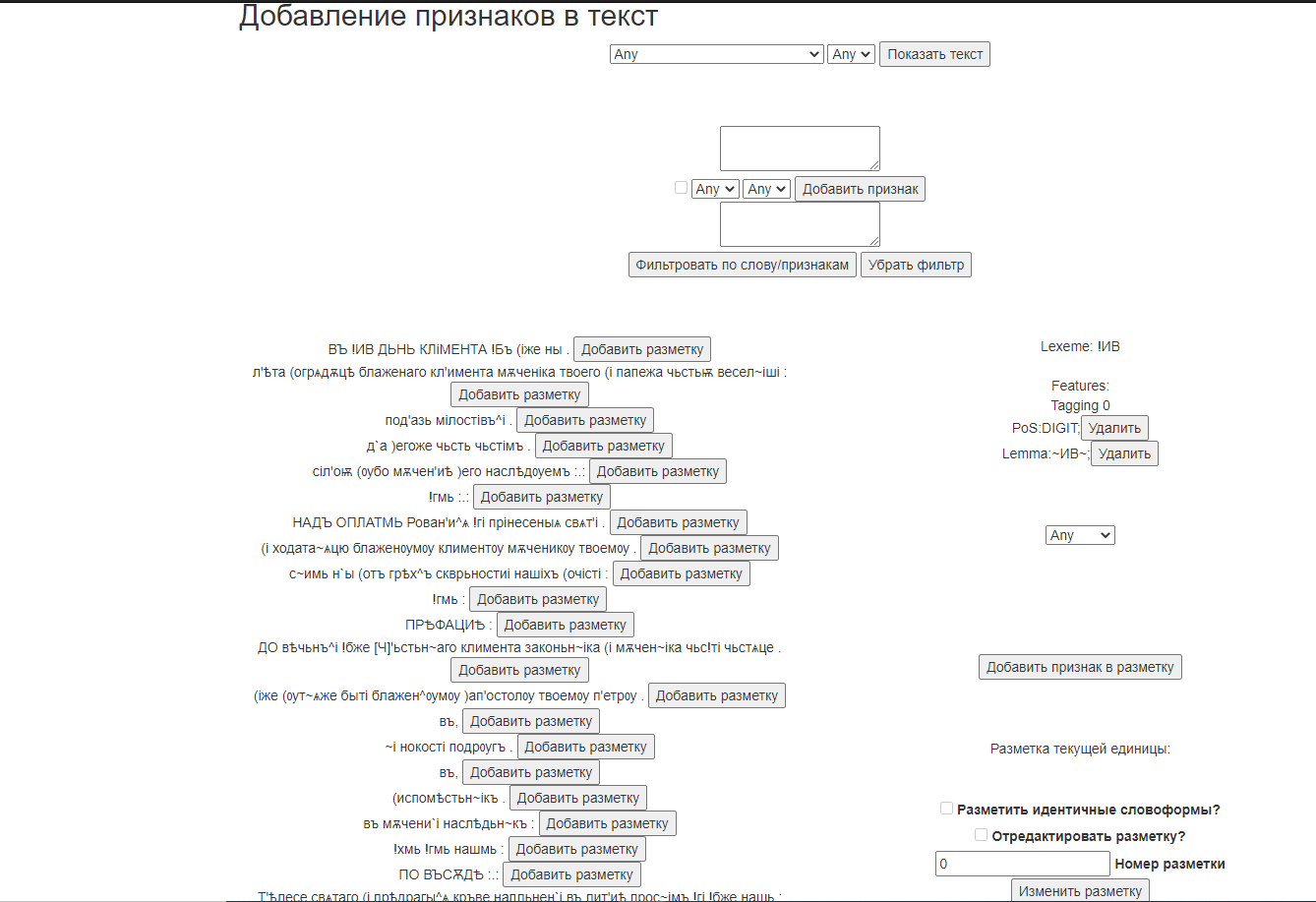


Рис 5. Окно подмодуля аннотирования текста.

Подмодуль визуализации текстовых особенностей предоставляет возможность графически отобразить созданную аннотацию выбранного подраздела выбранного памятника по сегментам, токенам и графемам. Для него так же, как и для подмодуля аннотирования, доступна функция отбора токенов по комбинации графем внутри них.

На данный момент единственный способ визуализации текстовых особенностей – выделение цветом единиц, которые соответствуют заданным пользователям критериям, а именно, наличию или отсутствию определённых признаков. Пример (выделение цветом только числительных) представлен ниже.

ВЪ ~ИВ~ ДЬНЬ КЛiМЕНТА ~Бъ iже ны .

Основной функционал корпуса реализован в подмодуле поиска. На данном этапе исследования возможен поиск по отдельным памятникам; формирование подкорпусов относится к будущим стадиям разработки системы. Помимо этого, пользователь может осуществить сортировку по частям токенов, а также заранее составленной аннотации памятников. Допустимо использование аннотации отдельных памятников, их разделов, сегментов, токенов и графем.

В выдачу включаются все соответствующие заданным параметрам токены. Отображение их на данный момент осуществляется без пагинации, все контексты выводятся на единственную страницу. Каждый токен выводится в формате *KWIC (Key Word In Context)*, «ключевое слово в контексте». В первой строке каждой единицы выдачи указываются идентификационный номер памятника, сам памятник и конкретный его раздел. Следующие строки занимают токены сегмента, идущие до ключевого слова. Затем, на отдельной строке, размещается само ключевое слово. За ним – оставшаяся часть сегмента.

Пример единицы выдачи представлен ниже. Он представляет собой фрагмент ответа на запрос на получение всех лемм в корпусе, начинающихся с ~.

7. 4\_ZogrB\_tagged: 4\_ZogrB\_tagged  
ишъдъ же рабъ тъ обрѣте единого отъ клеврѣтъ своихъ iже бѣ длъжънъ емѹ

**~р~**

пѣнѧsъ .

* 1. Выводы.

Корпус старославянского языка программно представлен приложением WinForms, написанным на C#, осуществляющим предварительную обработку текста, подключающим скрипты, написааные на Python, для частеречной разметки и лемматизации, а также приложением ASP.Net Core 2.1, являющимся реализацией универсальной системы создания корпусов для старославянского языка.

Дальнейшие улучшения программного представления корпуса старославянского языка будут связаны с развитием данной системы.

## **Заключение**

В работе был выявлен ряд ключевых лингвистических проблем, с которыми исследователь может столкнуться при исследовании старославянского языка, а также определены методы, которыми возможно данные проблемы решить.

В первую очередь, были охарактеризованы существующие корпусы старославянского языка. Ни один из них в полной мере не удовлетворяет критериям полноты представления старославянского языка. Возникла необходимость создания собственного корпуса.

Следует заметить, что выработка методологии создания корпуса с нуля выглядит тяжёлой и иррелевантной задачей. Корпусы для крупных индоевропейских языков создавались сушественно раньше, с использованием совершенно иных технологий, нежели тех, что доступны исследователям сейчас. Представляется возможным обратиться к опыту создания корпусов неиндоевропейских языков, в частности, арабского языка и языка паэс.

В работе предпринята попытка краткой характеристики старославянского языка как языкового идиома, отличного от других славянских языков. Представлены основные особенности, по которым возможно идентифицировать старославянские тексты, среди которых – особый рефлекс праславянского сочетания \**tj/*\**dj*.

Полностью определить текстовый состав корпуса старославянского языка не удалось: два текста во время написания соответствующих разделов работы находились вне доступа исследователей. Поставлен вопрос о языковой принадлежности ряда текстов, считающихся определяющими для старославянского канона (прежде всего – Киевских листков). Некоторые тексты (к примеру, Остромирово евангелие) были исключены из массива старославянских текстов в силу радикального отличия их языковых особенностей от языковых особенностей, указанных исследователями для старославянского языка как языкового идиома. Ещё одна группа текстов (например, Преславская надпись) на данный момент не может быть определена ни как тексты старославянского языка, ни как тексты какого-то другого идиома. Провести их анализ по существующим критериям в данный момент невозможно. Требуется разработка дополнительных критериев, что будет сделано в последующих исследованиях.

Старославянские тексты представлены в нескольких системах письменности. Для упрощения частеречной разметки и лемматизации потребовалось решить задачу унификации, приведения к кириллическому написанию глаголических рукописей, а также рукописей, переведённых исследователями в ASCII-кодировку. Основным методом решения этой задачи стали регулярные выражения.

Были проанализированы существующие подходы к токенизации, частеречной разметке и лемматизации.

В практической части токенизация осуществлялась путём разделения предварительно обработанного текста по пробельным символам. Это нарушило построчную схему представления текста, однако позволило дать более точное лингвистическое его представление.

Был создан расширенный набор тэгов для наиболее адекватного представления частеречного состава старославянского языка. Проведена частеречная разметка моделью, структурно схожей с TreeTagger, демонстрирующей результат в 81% точности на Мариинском евангелии, а также наибольшую надёжность на гетерогенном массиве текста.

Выявлено, что скрытая марковская модель, усиленная *n*-граммной, способна к относительно высокой генерализации. На Киевских листках она продемонстрировала преимущество над более современными методами, в частности, рекуррентными нейросетями. Точность *zero-shot* разметки (разметки «с нуля») достигла значения в 51%. При улучшении модели и увеличении тренировочных данных ожидается повышение как этих конкретных результатов, так и улучшение способности модели к генерализации.

Лемматизация была выполнена рекуррентной нейронной сетью.

Точность модели, при помощи которой осуществлялась лемматизация, на тестовом наборе данных из Мариинского евангелия достигла 85%. Этот показатель оказался меньше, чем у моделей, обученных на нескольких десятках языков, представленных в наборах данных UD, однако гораздо более важным оказалось то, что, будучи оснащённой дополнительной системой правил, эта модель показала высокую степень способности к генерализации на гетерогенных массивах текстов.

Для подсчёта точности были привлечены и дополнительные метрики, меры сходства строк, а именно расстояния Левенштейна, Дамерау-Левенштейна и Джаро-Винклера. Данный подход в лингвистике для оценки эффективности моделей, осуществляющих преобразование «последовательность-в-последовательность», ранее практически не применялся. Он позволил более точно оценить то, насколько результаты, показанные моделью, посимвольно отличались от золотого стандарта. Помимо этого, при помощи выбросов, обнаруженных по данным метрикам, удалось идентифицировать источник неточностей модели, а именно – её склонность к генерации коротких последовательностей.

Точность модели на Киевских листках приблизилась к 50–55%. Основываясь на этих данных, видится возможным утверждать, что была доказана её способность к генерализации. Модель с дальнейшими усовершенствованиями будет применена для разметки и иных текстов старославянского языка.

И модуль частеречной разметки, и модуль лемматизации предстоит улучшать в дальнейшем, возможно, с применением новых технологий и новых архитектур нейронных сетей.

Результаты работы представлены в формате .json-файлов, полученных через написанное на C# приложение с подключаемыми модулями на Python, отображаемых через универсальную систему представления текстовых корпусов, написанную на C#/JavaScript/Python. К настоящему моменту обработаны и автоматически размечены тексты Ассеманиева, Мариинского, Зографского (включая рукопись Б) евангелий, Супрасльской рукописи, Саввиной книги и Киевских листков. Данная разметка сейчас корректируется вручную. Объём корпуса будет увеличиваться за счёт включения иных старославянских рукописей. В данный момент работа ведётся над текстом Енинского апостола.

Исследовательская гипотеза может считаться доказанной: несмотря на высокую степень гетерогенности, старославянский язык может быть обработан и помещён в корпус как единый идиом.

## **Список использованной литературы**

1. *Алпатов В. М.* О разных подходах к выделению частей речи // Вопросы языкознания. — М.: ИРЯ РАН, 1986. — Вып. 4. — С. 37–46.
2. *Афанасьев И. А.* Корпус старославянского языка: недостающее звено в диахронической славистике // Slavica iuvenum XXI : sborník příspěvků z mezinárodní vědecké konference Slavica iuvenum 2020, 31.3. a 1.4.2020. – Ostrava : Ostravská univerzita, 2020. – С. 13 – 22.
3. *Афанасьева Т. И., Козак В. В., Соболев А. Н.* Глаголическая письменность Западных Балкан X–XVI веков: учебно-методическое пособие. СПб.: Наука, 2016. — 71 с.
4. *Баранов В. А.* К вопросу об использовании статистических методов для поиска коллокаций и коллигаций в древнейших славянских текстах (на материале глаголических рукописей корпуса «Манускрипт») // SLOVO. – Zagreb, Croatia: Staroslavenski institut. – 2019. – Вып. 69 – С. 1 – 33.
5. *БФССЯ –* Большой фразеологический слофарь старославянского языка / гл. ред. С. Г. Шулежкова; ред. колл.: С. А. Анохина, А. А. Осипова, Н. В. Позднякова. – Магнитогорск: Научно-исследовательская словарная лаборатория МГТУ им. Г. И. Носова, 2021. – Т. 1. – 555 с.
6. *Вайан А.* Руководство по старославянскому языку. – М.: Издательство иностранной литературы, 1952. – 447 с.
7. *Гранстрем Е. Э.* О происхождении глаголической азбуки. // ТОДРЛ. – СПб: Отдел древнерусской литературы ИРЛ (Пушкинский Дом) РАН, 1955. – С. 300 – 317.
8. *Гришина Е. А.* Мультимедийный русский корпус (МУРКО): проблемы аннотации // Национальный корпус русского языка: 2006—2008. Новые результаты и перспективы. – СПб.: Нестор-История, 2009. – С. 175—214.
9. *Добрев И.* Палимпсестовите части на Зографското Евангелие. *//* Сб. «Константин-Кирил Философ. Доклади от симпозиума, посветен на 1100-годишнината от смъртта му». – София: Изд-во на Българската акад. на науките, 1971. – С. 157—164.
10. *Ильинский Г. А.* Македонский листок. – СПб.: Имперская Академия Наук, 1906. – 26 с.
11. *Карский Е. Ф.* Славянская кирилловская палеография. – М.: Наука, 1979. – 494 с.
12. *Крашенинникова О. А.* Древнеславянский Октоих св. Климента, архиепископа Охридского: По древнерусским и южнославянским спискам XIII—XV веков. – М.: Языки славянских культур, 2006. — 384 с., ил. — (Studia philologica).
13. *Крючкова О. Ю., Гольдин В. Е.* Саратовский диалектный корпус: новый научный и образовательный ресурс. Концепция, методические материалы. – Саратов: ИД «Научная книга», 2010. – 35 с.
14. *Медынцева А. А., Попконстантинов К.* Надписи из Круглой церкви в Преславе. — София: Болгарская академия наук, 1985. — 131 с.
15. *Мейе А.* Общеславянский язык. – М.: Издательская группа «Прогресс», 1934. – 499 с.
16. *Миклас Х.*, *Садовски В., Хюрнер Д., Вандл Ф.* Синайский глаголический Литургиарий («Миссал»). Эл. рес.: http://manuscripts.ru/mns/portal.main?p1=67
17. *Мирчев К., Кодов Х.* Енински апостол. Старобългарски паметник от XI в. – София*:* Изд-во на Бълг. акад. на науките*,* 1965*.* – 263 с.
18. *ОРД* – Корпус спонтанной речи носителей русского языка «Один речевой день». Эл. рес.: http://ord-corpus.spbu.ru/megapolis/
19. *Поливанова А. К.* Старославянский язык. Грамматика. Словари. – М.: Университет Дмитрия Пожарского, 2013. – 792 с.
20. *Селищев А. М.* Старославянский язык. Часть первая. Введение. Фонетика. – М.: Государственное учебно-педагогическое издательство министерства просвещения РСФСР, 1951. – 336 с.
21. *Соболевский А.И.* Материалы и исследования в области славянской филологии и археологии. – СПб.: Императорская Академия Наук, 1910. – 286 с.
22. *Срезневский И. И.* Древние глаголические памятники сравнительно с памятниками кириллицы. – СПб.: Императорская Академия Наук, 1866. – 314 с.
23. *СС –* Старославянский словарь (по рукописям X-XI вв.) Ок. 10000 сл. / ред. Р. М. Цейтлин, ред. Р. Вечерка, ред. Э. Благова. – М. : Рус. яз., 1994. – 842 с.
24. *Успенский Ф.* *И.* Надпись царя Самуила // Извѣстія Русскаго Археологическаго института въ Константинополѣ. — Софія : Държавна печатница. – 1899. — Т. 4. – Вып. 1. – С. 1 – 4.
25. *Шафарик П.Й.* О происхождении и родине глаголитизма // Чтения Общества истории и древностей Российских. – М.: Московский университет, 1860. – Т. 4. – Ч. 3. – С. 1 – 66.
26. *Abumalloh R. A., Al-Sarhan H. M., Ibrahim O. B., Abu-Ulbeh W.* Arabic Part-of-Speech Tagging // Journal of Soft Computing and Decision Support Systems. – Skudai: Penerbit UTM Press, 2016. – Т. 3. – № 2. – С. 45 – 52.
27. *Aduriz I., Aldezabal I., Artola X., Ezeiza N., Urizar R.* EUSLEM: A lemmatiser/tagger for Basque // Proceedings of the 7th EURALEX International Congress, Novum Grafiska AB, Göteborg, Sweden. – Luxembourg: Springer. – 1998. – С. 27 – 35.
28. *Afanasev I.* Corpus-dictionary system: introducing a concept // International conference “Linguistic Forum 2020: Language and Artificial Intelligence”. 12–14 November 2020. Institute of Linguistics RAS, Moscow, Russia: Abstracts / Andrej A. Kibrik, Valentin Gusev, Dmitry Zalmanov (ред.). — 164 с. — Moscow: Institute of linguistics RAS, 2020. – С. 10 –11.
29. Akhmetov I., Pak A., Ualiyeva I., Gelbukh A. Highly Language-Independent Word Lemmatization Using a Machine-Learning Classifier // Computacion y Sistemas. – Mexico City, Mexico: IPN. – 2020. – Т. 24. – С. 1353 – 1364.
30. *Alfaifi A., Atwell E.* A*.* Potential Uses of the Arabic Learner Corpus // Leeds Language, Linguistics and Translation PGR Conference. – Leeds: University of Leeds. – 2013. – С. 1 – 3.
31. *Alfaifi A., Atwell E.* B*.* Arabic Learner Corpus v1: A New Resource for Arabic Language Research // Second Workshop on Arabic Corpus Linguistics (WACL-2). – Lancaster: University of Lancaster. – 2013. – С. 10 – 12.
32. *Alfaifi A., Atwell E.* С. Arabic Learner Corpus: Texts Transcriptions and Files Format // Корпусная лингвистика – 2013. – Санкт-Петербург: Санкт-Петербургский государственный университет. – 2013. – С. 13 – 20.
33. *Alrabia M.*, *Al-Salman A., Atwell E., Alhelewh N.* KSUCCA: A Key To Exploring Arabic Historical Linguistics // International Journal of Computational Linguistics. – Kuala Lumpur: CSC Journals. – 2014. – Т. 2. – Ч. 1. – С. 27 – 36.
34. *Attia, M.A.* Arabic Tokenization System // Proceedings of the 5th Workshop on Important Universal Matters. – Madison: Omnipress. – 2007. – 120 с. – С. 65 – 72.
35. *Atwell E., Al-Sulaiti L., Ai-Osaimi S., Shawar B.* A review of Arabic corpus analysis tools – Un Examen d'Outils pour l'Analyse de Corpus Arabes. // Proceedings of TALN04: XI Conference sur le Traitement Automatique des Langues Naturelles. – Marseille. – 2004. – Т. 2. – Ч. 2. – 234 с. – С. 229 – 234.
36. *Atwell E.*, *Al-Sulaiti L.* The design of a corpus of Contemporary Arabic // International Journal of Corpus Linguistics. – Amsterdam: John Benjamins Publishing Company. – 2006. – Т. 11. – Вып. 2. – 140 с. – С. 135 –171.
37. *Atwell E.* Development of tag sets for part-of-speech-tagging // Corpus Linguistcs: An International Handbook. – Berlin. NY: Walter de Gruyter. – 2008. – 1353 с. – С. 501 – 524
38. *Behera P.* An Experiment with the CRF++ Parts of Speech (POS) Tagger for Odia // Language in India. – Bloomington: M. S. Thirumalai. – 2017. – Т. 17. – Вып. 1. – 312 с. –18 – 40.
39. *Berdičevskis A.* The beginning of a beautiful friendship: rule-based and statistical analysis of Middle Russian // Komp'yuternaya lingvistika i intellektual'nye tekhnologii. Trudy mezhdunarodnoj konferencii «Dialog». – Moscow, Russia: RSSU. – 2016. – С . 99 – 111.
40. *Bergmanis T., Goldwater S.* Context sensitive neural lemmatization with Lematus // Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). – New Orleans, Louisiana: ACL. – 2018. – Т. 1. – С. 1391 – 1400.
41. *Bird, S., Loper, E., Klein, E.* Natural Language Processing with Python. – Sebastopol: O’Reilly Media Inc., 2009. – 504 с.
42. *Branco A.*, *Silva J.* Portuguese Specific Issues in the Rapid Development of State of the Art Taggers // Tagging and Shallow Processing of Portuguese: workshop notes of TASHA’2003. – Lisbon: University of Lisbon. – 2003. – 38 с. – С. 7 – 9.
43. *Camps J.-B., Gabay S., Fièvre P., Clérice T., Cafiero F*. Corpus and Models for Lemmatisation and POS-tagging of Classical French Theatre. Эл. рес.: https://arxiv.org/pdf/2005.07505.pdf
44. *Cartoni B.* Introducing the MuLexFoR : A Multilingual Lexeme Formation Rule Database // eLEX-2009. Book of abstracts. eLexicography in the 21st century: New challenges, new applications. – Lowain-la-Neuve: Centre for English Corpus Linguistics, Université catholique de Louvain. – 2009. – С. 39 – 40.
45. *Carstens A.*, *Eiselen A.* Designing a South African Multilangual Learner Corpus of Academic Texts (*SAMuLCAT)* // Language Matters. – Cape Town: Taylor & Francis. – 2019. – Т.50. – Вып. 1. – 136 с. – С. 64 – 83.
46. *CCMH* – Corpus Cyrillo-Methodianum Helisingiense. Эл.рес.: http://www.helsinki.fi/slaavilaiset/ccmh/index.html
47. *Chiarcos C., Fiedler I., Grubic M., Haida A., Hartmann K,, Ritz J., Schwarz A., Zeides A., Zimmermann M.* Information Structure in African Languages: Corpora and Tools // Proceedings of the EACL 2009 Workshop on Langauge Technologies for African Languages – AfLaT 2009. – Athens: Tehnografia Digital Press. – 2009. – 131 c. – С. 17 – 24.
48. *Cho K., Merrienboer B. V., Gülçehr Ç., Bahdana D*., *Bougares F*., *Schwenk H., Bengio Y.* Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). –Doha, Quatar: ACL. – С. 1724 – 1734.
49. *Chrupała G*. Simple data-driven context-sensitive lemmatization // Procesamiento del Lenguaje Natural. – Spain: Sociedad Española para el Procesamiento del Lenguaje Natural. – 2006. – Вып. 37. – С. 121 – 127.
50. *Cyrillomethodiana –* Исторический корпус болгарского языка. Эл. рес.: https://histdict.uni-sofia.bg/
51. *Cyrillic Unicode.* Эл. рес.: https://github.com/slavonic/cu-tex
52. *Damerau F. J*. A technique for computer detection and correction of spelling errors // Communications of the ACM. – New York: ACM. – 1964. – Т. 7. – Вып. 3. – С. 171 – 176.
53. *Dandapat S.*, *Sarkar S., Basu A.* Automatic Part-of-Speech Tagging for Bengali: An Approach for Morphologically Rich Languages in a Poor Resource Scenario // Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions. – Prague: Association for Computational Linguistics, 2007. – 228 с. – С. 221 – 224.
54. *Danso S.*, *Lamb W.* Developing an Automatic Part-of-Speech Tagger for Scottish Gaelic // Proceedings of the First Celtic Language Technology Workshop. – Dublin: Association for Computational Linguistics. – 2014. – 80 с. – С. 1 – 5.
55. *Dash, N.S., Chaudhuri, B.B.* Why do we need to develop corpora in Indian Languages? // Proceedings of International Conference on SCALLA, Bangalore (2001). – Bangalore, India: Indian Institute of Science. – 2001. – С. 1 – 18.
56. *Daud A., Khan W., Che D*. Urdu language processing: a survey // Artificial Intelligence Review. – Luxembourg: Kluwer Academic Publishers. – 2017. – Т. 47. – Вып. 3. – 311 с. – С. 279– 311.
57. *Deepali G. K., Ramesh N.R., Namrata C. M.* Rule Based Part-of-Speech Tagger for Marathi Language // International Journal of Scientific Research and Technology. – Gujarat: Gandhinagar Technoscience Academy. – Т. 4. – Вып. 5. – 1935 с. – С. 1607 – 1612.
58. *Dhonnchadha E. U., Frenda A., Vaughan B.* Issues in Designing a Spoken Corpus of Irish // Proceedings of the Workshop on Language Technology for Normalisation of Less-Resourced Languages (SALTMIL8/AfLaT2012). – Paris: European Language Resources Association, 2012. – С. 1 – 6.
59. *Divjak D., Sharoff S., Erjavec T.* Slavic Corpus and Computational Linguistics // Journal of Slavic Linguistics. – Indiana: Slavica Publishers. – 2017. – Т. 25. – Вып. 2. – С. 171 –198.
60. *Džeroski S., Erjavec T.* Learning to Lemmatise Slovene Word // Cussens J., Džeroski S. (eds.) Learning Language in Logic. LLL 1999. Lecture Notes in Computer Science. – Berlin, Heidelberg: Springer. – 2000. – Т. 1925. – С. 69 – 88.
61. *Eckhoff H. M., Berdicevskis A.* Linguistics vs. digital editions: The Tromsø Old Russian and OCS Treebank // Scripta & e-Scripta. – София: БАН, 2015. – Т. 14 – Вып. 15. – С. 9 – 25.
62. *Egbert J., Larsson T., Biber D.* Doing linguistics with a corpus. – Cambridge University Press: Cambridge, 2020. – 88 с.
63. *Elenius K., Forsbom E., Megyesi B*. Siurvey on Swedish Language Resources. – Uppsala: Uppsala University, 2008. – 90 с.
64. *Evans R., Brown D., Corbett G., Tiberius C.* Russian Lemmatisation with DATR. – University of Brighton: Brighton, 2006. – 12 c.
65. *Farkas R., Vincze V., T I., Ormándi R., Szarvas G., Almási A.* "Web-Based Lem-matisation of Named Entities // Proceedings of TSD 2008, Brno, Czech Republic. – Luxembourg: Springer. – 2008. – С. 53 – 60.
66. *Fernández L.* A contribution to Old English lexicography: Utgangan, wiðhealdan, ofersceadan, onbefeallan and ongangan // NOWELE. North-Western European Language Evolution. – Amsterdam, the Netherlands: John Benjamins Publishing Company. – 2020. – Т. 73. – Вып. 2. – С. 236 – 251.
67. *Freeman S., Herron J. C.* Evolutionary Analysis (3rd ed.). – Upper Saddle River, NJ: Pearson Education, 2004. – 816 c.
68. *FuzzyWuzzy* ­– Эл. рес.: https://github.com/seatgeek/fuzzywuzzy
69. *Gambäck B., Asker L.* Experiences with developing language processing tools and corpora for Amharic // Cunningham, P., Cunningham, M. (eds.) Proceedings of IST-Africa 2010, the 5th Conference on Regional Impact of Information Society Technologies in Africa. – Dublin, Ireland: IIMC. – 2010. – С. 1 – 8.
70. *Gambäck B., Olsson F., Argaw A. A., Asker L.* Methods for Amharic Part-of-Speech Tagging // Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009. – Athens: Association for Computational Linguistics. – 2009. – 131 с.– С. 104 – 111.
71. *Gambäck B.* Tagging and Verifying an Amharic News Corpus // Proceedings of the Workshop on Language Technology for Normalisation of Less-Resourced Languages (SALTMIL8/AfLaT2012). – Paris: European Language Resources Association, 2012. – С. 79 – 84.
72. *Gesmundo A., Samardžić T*. Lemmatisation as a tagging task // Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. – Jeju Island, Korea: ACL. – 2012. – Т. 2. – С. 368 – 372.
73. *Glibetić N.* A New Eleventh-Century Glagolitic Fragment from St Catherine's Monastery: The Midnight Prayer of Early Slavic Monks in the Sinai *//* Археографски прилози. – Београд: Нар. библ. Србиjе. Археографско од-ње, 2015. – Вып. 37. – С. 11 – 47.
74. *Grivec F., Tomšić F.* Constantinus et Methodius Thessalonicenses, Fontes // Radovi Staroslavenskog instituta. – Zagreb: Staroslavenski institut. – 1960. – Т. 4. – Вып. 4. – С. 13 – 276.
75. *Groenewald H.* Automatic lemmatisation for Afrikaans. – North-West University, Potchefstroom Campus: SAR*,* 2007. – 22 с.
76. *Groenewald H.* Using technology transfer to advance automatic lemmatisation for Setswana // Proceedings of the First Workshop on Language Technologies for African Languages. – Stroudsburg, PA: ACL. – 2009. – С. 32 – 37.
77. *Gouws R.H., Prinsloo D.* Lemmatisation of Adjectives in Sepedi // Lexikos. – SAR: Bureau WAT. – 2012. – Т. 7. – С. 45 – 57.
78. *Grubbs F. E.* Procedures for detecting outlying observations in samples // Technometrics. – Boston, Massachusetts: ASA & ASQA. – 1969. – Т. 11. – Вып. 1. – С. 1 – 21.
79. *Hann M.* Principles of Automatic Lemmatisation // *ITL Review of Applied Linguistics*. – Amsterdam: John Benjamins Publishing Company. – 1974. – Т. 23. – Вып. 1. – С. 3 – 22.
80. *Hardie A., Lohani R., Yadava Y.* Extending corpus annotation of Nepali: advances in tokenisation and lemmatization // Himalayan Linguistics. – 2014. – Т. 10. – Вып. 1. – С. 151 – 165.
81. *Helmut S.* Probabilistic Part-of-Speech Tagging Using Decision Trees // Proceedings of International Conference on New Methods in Language Processing. – Manchester, UK: ACL. – 1994. – С. 1–9. Эл. Рес.: https://www.cis.unimuenchen.de/~schmid/tools/TreeTagger/data/tree-tagger2.pdf
82. *HMM*. Эл. рес.: https://github.com/The-One-Who-Speaks-and-Depicts/hmm-pos-tagger
83. *Jaro M. A.* Advances in record linkage methodology as applied to the 1985 census of Tampa Florida // Journal of the American Statistical Association. – Boston, Massachusets: ASA. – 1989. – Т. 84. – С. 414 – 420
84. *Jiampojamarn S., Cherry C., Kondrak G.* Joint processing and discriminative training for letter-to-phoneme conversion // Proceedings of ACL-08: HLT. – Columbus, Ohio: ACL. – 2008. – С. 905 – 913.
85. *Jurish B.* A Hybrid Approach to Part-of-Speech Tagging. – Berlin: Berlin-Brandenburgishe Akademie der Wissenschaften, 2003. – 27 с.
86. *Jursic M., Mozetic I., Erjavec T., Lavrac N.* LemmaGen: Multilingual Lemmatisation with Induced Ripple-Down Rules // JUCS. – Austria: Austria Competence Center for Knowledge Manadgement. – 2010. – Т. 16. – С. 1190 – 1214.
87. *Ismail O., Yagi S., Hammo B.* Corpus Linguistic Tools for Historical Semantics in Arabic // International Journal of Arabic-English Studies (IJAES). – Keserwan: Librairie du Liban Publishers, 2014. – Вып. 15. – C. 135 – 152.
88. *Kamphuis J.* Verbal Aspect in Old Church Slavonic. – Brill: Leiden, 2020. – 329 с.
89. *Kanerva J., Ginter F., Salakoski T.* Universal Lemmatizer: A sequence-to-sequence model for lemmatizing Universal Dependencies treebanks // Natural Language Engineering. – Cambridge, UK: Cambridge University Press. – 2020. – С. 1 – 30.
90. *Kempgen S.* *S.* Unicode 5.1, Old Church Slavonic, Remaining Problems – and Solutions, including OpenType Features // Slovo: Towards a Digital Library of South Slavic Manuscripts. Proceedings of the International Conference, 21 – 26 February 2008. – Sofia, Bulgaria: Sofia. – 2008. – С. 200 – 219.
91. *Khalifa S., Habash N., Eryani F., Obeid Os., Abdulrahim D., Al Kaabi M.* A Morphologically Annotated Corpus of Emirati Arabic // Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). – 2018. – С. 3839 – 3846.
92. *Koeva S*., *Leseva S., Rizov B., Tarpomanova E., Dimitrova T., Kukova H., Todorova M.* Design and development of the Bulgarian sense-annotated corpus // Actas del III Congreso Internacional de Lingüística de Corpus. Las Technologiás de la Información y las Comunicaciones: Presente y Futuro en el Análisis de Corpus. – Valencia, Spain: Universitat Politèchnica de València. – 2011. – С. 143 – 150.
93. *Koseska-Toszewa V., Roszko R.* Remarks on Classification of Part of Speech and Classifiers in an Electronic Dictionary // Proceedings of MONDILEX First Open Workshop “Lexicographic Tools and Techniques”. – Moscow, Russia: IITP RAS. – 2008. – С. 80 – 88.
94. *Kosch I.M*. Lemmatisation of Fixed Expressions: The Case of Proverbs in Northern Sotho // Lexikos. – SAR: Bureau WAT. – 2016. – Т. 26. – С. 145 – 161*.*
95. *Kumar S. S., Kumar M. A., Soman K.P.* Experimental analysis of Malayalam PoS tagger using EPIC framework in Scala // ARPN Journal of Engineering and Applied Sciences. – Islamabad: Asian Research Publishing Network, 2016. – Т. 11. – Вып. 13. – С. 8017 – 8023.
96. *Lemmatiser.* Эл. рес.: https://github.com/The-One-Who-Speaks-and-Depicts/OCS-corpus-lemmatiser
97. *Levenshtein V. I.* Binary codes capable of correcting deletions, insertions, and reversal // Soviet Physics Doklady. – M.: MAIK Nauka/ Interperiodica and Springet Science+Business Media. – 1966. – Т. 10. – Вып. 8. – С. 707 – 710.
98. *Ljubešić N., Dobrovoljc K.* What does Neural Bring? Analysing Improvements in Morphosyntactic Annotation and Lemmatisation of Slovenian, Croatian and Serbian // Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing. – Florence, Italy: ACL. – 2019. – С. 29 – 34.
99. *Loftsson H.*, *Kramarczyk I., Helgadóttir S., Rögnvaldsson E.* Improving the PoS tagging accuracy of Icelandic text // Proceedings of the 17th Nordic Conference of Computational linguistics NODALIDA 2009. – Odense: Northern European Association for Language Technology (NEALT). – 2009. – 236 с. – С. 103 – 110.
100. *Loftsson H. T*agging Icelandic text: A linguistic rule-based approach // Nordic Journal of Linguistics. – Cambridge: Cambridge University Press. – 2008. – Т. 31. – Вып. 1. – 128 с. – С. 47 – 72.
101. *Mansour M.A.* The Absence of Arabic Corpus Linguistics: A Call for Creating an Arabic National Corpus. // International Journal of Humanities and Social Science. – Los Angeles. – 2013. – Т. 3. – Вып. 12. – С. 81 – 90.
102. *Martinez L.M. S., Cobos A. C., Muňos J. C. C., Curieux T. R., Herrera-Viedma E., Peluffo-Ordoňez D. H.* Building a Nasa Yuwe Language Corpus and Tagging with a Metaheuristic Approach // Computación y Sistemas. – Mexico: Centro de Investigación en Computación. – 2018. – Т. 4. – Вып. 3. – С. 881 – 894.
103. *MacKenzie C.E.* Coded Character Sets: History and Development. –Reading, Mass: Addison-Wesley, 1979. – 513 с.
104. *Miklas H., Sadovski V.* Die Struktur des Altkirchenslavishen // Die slavishen Sprachen. – Berlin: De Gruyter Moyton. – 2014. – Т. 2. – С. 1252 – 1275.
105. *Milintsevich K., Sirts K*. Lexicon-Enhanced Neural Lemmatization for Estonian // Proceedings of the Human Language Technologies – The Baltic Perspective. – Amsterdam: IOS Press. – 2020. – С. 158 – 165.
106. *Mills J*. Lemmatisation of the Corpus of Cornish // Proceedings of the Workshop on Language Resources for European Minority Languages. – Granada, Spain: LREC First International Conference on Language Resources and Evaluation. – 1998. – С. 1 – 6.
107. *Mirzanezhad Z*., *Feizi-Derakhshi M.-R.* Using morphological analyzer to statistical POS Tagging on Persian Text // IJCSIS. – USA: IJCSIS. – 2016. – Т. 14. – Вып. 8. – С. 1093 – 1103.
108. *Mohamed Elhadj Y.* *O.* Statistical Part-of-Speech Tagger for Traditional Arabic Texts // Journal of Computer Science. – NY: Science Publications. – 2009. – Т. 5. – Вып. 11. – 1100 с. – С. 794 – 800.
109. *Mohan Raj S, N.*, *Rajendran S.* Tamil oriented machine translation systems under Indian Language to Indian Language Machine translation (ILILMT) consortium // Proceedings of the International Tamil Internet Conference. – Эл.рес.: https://www.researchgate.net/publication/329962105\_Tamil\_oriented\_machine\_translation\_systems\_under\_Indian\_Language\_to\_Indian\_Language\_Machine\_translation\_ILILMT\_consortium
110. *Mulhall C.* The Lemmatisation of Lexically Variable Idioms: The Case of Italian-English Dictionary // Proceedings of the 13th EURALEX International Congress, Barcelona, Spain. – Luxembourg: Springer. – 2008. – С. 1373 –1378.
111. *Mzamo L., Helberg A., Bosch S.* Introducing XGL - a lexicalised probabilistic graphical lemmatiser for isiXhosa // Proceedings of 2015 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech). – Port Elizabeth: PRASA. – 2015. – С. 142 – 147.
112. *Netzer Y., Adler M., Gabay D., Elhadad M.* Can You Tag the Modal? You Should. // Proceedings of the 5th Workshop on Important Universal Matters. – Madison: Omnipress. – 2007. – 120 с. – С. 57 – 64.
113. *Nthambeleni M., Musehane N*. The Lemmatisation of Nouns in Tshivenda Dictionaries // Lexikos. – SAR: Bureau WAT. – 2014. – Т. 24. – С. 214 – 224.
114. *Paliga S.* Working With Old Church Slavonuc Texts: the Simple Way from Non-Standard Encoding to Unicode Encoding // Romanoslavica. – Bucharest, Romania: Romanian Association of Slavic Studies. – 2018. – Т. 4. – Вып. 1. – С. 65 – 77.
115. *Plisson J., Lavrac N., Mladenić D., Erjavec T*. Ripple Down Rule learning for automated word lemmatization // AI Communications. – Amsterdam: IOS Press. – 2008. – Т. 21. – С. 15 – 26.
116. *Podtergera I*. SlaVaComp-Lemmatizer: a Lemmatization Tool for Church Slavonic // Proceedings of El’Manuscript-2016: Textual Heritage and Information Technologies. – Izhevsk: ISU. – 2016. – С. 212 – 221.
117. *Preprocessor*. Эл.рес.: https://github.com/The-One-Who-Speaks-and-Depicts/old\_slavic\_process
118. *Prinsloo D.* A Critical Analysis of the Lemmatisation of Nouns and Verbs in isiZulu // Lexikos. – SAR: Bureau WAT. – 2012. – Т. 21. – С. 169 – 193.
119. *Radziszewski A.* Learning to lemmatise Polish noun phrases // Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. – Sofia, Bulgaria: ACL. – 2013. – Т. 1. – С. 701 – 709.
120. *Rajendran S., Krishnakumar K.* A Comprehensive Study of Shallow Parsing and Machine Translation in Malaylam. – Coimbatore: Amrita Vishwa Vidyapeetham, 2019. – 295 с.
121. *Reddy S., Sharoff S.* Cross Language POS Taggers (and other Tools) for Indian Langauges: An Experiment with Kannada using Telugu Resources // Proceedings of the 5th International Joint Conference on Natural Language Processing. – Chiang Mai: Asian Federation of Natural Languuage Processing. – 2011. – 1476 с. – С. 11 – 19.
122. *Romer U.* Corpis Research Applications in Second Language Teaching // Annual Review of Applied Linguistics. – Cambridge: Cambridge University Press. – 2011. – Т. 31. – С. 205 – 225.
123. *Saharia N., Das D., Sharma S., Kalita J.* Part of Speech Tagger for Assamese Text // Proceedings of the ACL-IJCNLP 2009 Conference Short Papers. – Singapore: World Scientific Publishing Co Pte Ltd. – 2009. – 393 с. – С. 33 – 36.
124. *Sánchez-Marco C., Boleda G., Padró L.* Extending the tool, or how to annotate historical language varieties // Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences and Humanities. – Madison: Omnipress. – 2011. – 145 с. – С. 1 – 9.
125. *Schryver G.-M. de, Nabirye M.* Corpus-driven Bantu Lexicography, Part 2: Lemmatisation and Rulers for Lusoga // Lexikos. – SAR: Bureau WAT. – 2018. – Т. 28. – С. 79 – 111.
126. *Simov K., Osenova P., Slavcheva M.* BTB-TR03: BulTreeBank. Morphosyntactic Tagset. – 2004. – Эл. рес.:  http://bultreebank.org/wpcontent/uploads/2017/06/BTB-TR03.pdf
127. *Singh T.D., Bandyopadhyay S.* Web Based Manipuri Corpus for Multiword NER and Reduplicated MWEs Identification using SVM // Proceedings of the 1st Workshop on South and southeast Asian Natural Language Processing. – Beijing, China: Chinese Information Processing Society of China. – 2010. – С. 35 – 42.
128. *Singha K.R., Purkayastha B. S., Singha K. D.* Part of Speech Tagging in Manipuri: A Rule-based Approach // International Journal of Computer Applications. – Geneva: Inderscience Publishers. – 2012. – 54 с. – С. 31 – 36.
129. *Spyns P.* A tagger/lemmatiser for Dutch medical language // Proceedings of the 16th conference on Computational linguistics. – Stroudsburg, PA: ACL. – 1996. – Т. 2. – С. 1147 – 1150.
130. *Straka M., Straková J.* Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe // Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. – ACL: Vancouver, Canada. – 2017. – С. 88 – 99.
131. *Straka M., Hajič J., Straková J.* UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing // Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Portorož, Slovenia. – Luxembourg : ELRA. – 2016. – С. 4290 – 4298.
132. *Streiter O., Stuflesser M., Ties I.* CLE, an aligned Tri-lingual Ladin-Italian-German Corpus, Corpus Design and Interface // Proceedings of SALTMIL Workshop at LREC 2004: First Steps in Language Documentation for Minority Languages. – Luxembourg: ELRA. – 2004. – С. 84 – 87.
133. *Strobl C., Malley J., Tutz G.* An Introduction to Recursive Partitioning: Rationale, Application and Characteristics of Classification and Regression Trees, Bagging and Random Forests // Psychological Methods. – Washington, DC: APA. – 2009. – Т. 14. – Вып. 4. – С. 323 – 348.
134. *Sutskever I., Vinyals O., Le Q. V.* Sequence to Sequence Learning with Neural Networks // NIPS. – NY, USA: ACM. – 2014. – С. 1 – 9.
135. *Tamburini F.* The AnIta-Lemmatiser: A Tool for Accurate Lemmatisation of Italian Texts // Proceedings of EVALITA 2012. – Rome, Italy: ACL. – 2013. – С. 266 – 273.
136. *Tarnanidis I.* The Psalter of Dimitri the Oltarnik // The Slavonic Manuscripts Discovered in 1975 at St. Catherine’s Monastery on Mount Sinai*. –* Thessaloniki: Hellenica Association for Slavic Studies. – 1988*. –* 366 с. – С. 91 — 100.
137. *Tianqi C., Carlos G*.XGBoost: A Scalable Tree Boosting System // Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. – San Francisco, CA, USA: ACM. – 2016. – С. 785–794.
138. *TITUS* – Thesaurus Indogermanisher Text- und Sprachmaterielen. Эл.рес.: http://titus.uni-frankfurt.de/indexe.htm
139. *Thompson, K.* Programming Techniques: Regular expression search algorithm // Communications of the ACM. – New York: ACM. – 1968. – Т. 11 – Вып. 6. – С. 419 – 422.
140. *TOROT*. Эл.рес.: https://nestor.uit.no/
141. *Tufiş D.* Algorithms and Data Design Issues for Basic NLP Tools // Language Engineering for Lesser-Studied Languages. – Amsterdam: IOS Press. –2009. – Т. 21. – 333 с. – С. 3 – 50.
142. *UD UPOS tag set.* Эл. рес.: https://universaldependencies.org/u/pos/
143. *Uludoğan G*. HMM PoS Tagger. Эл.рес.: https://github.com/gokceuludogan/hmm-pos-tagger.
144. *Unicode Standard.* Эл. рес.: http://www.unicode.org/charts/PDF/U2C00.pdf
145. *Wątróbska H., Kubiak R.* Wykorzystanie Emacsa, Haskella i TEXa w pracach nad słownikiem języka staro-cerkiewno-słowiańskiego // Biuletyn GUST. – Poland: GUST. – 2004. – Вып. 20. – С. 48 – 53.
146. *Winkler W. E.* String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage // Proceedings of the Section on Survey Research Methods. – Alexandria, VA: ASA. – 1990. – С. 354 – 359.
147. *Yousif J. H.* Natural Language Processing based Soft Computing // International Journal of Computer Applications. – Geneva: Inderscience Publishers. – 2013. – 49 с. – С. 43 – 49.
148. *Zeman D., Nivre J., Abrams M., et al.* Universal Dependencies 2.7, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. – 2020. – Эл.рес.: http://hdl.handle.net/11234/1-3424.
149. *Zeroual I., Lakhouaja A., Belanbib B.* Towards a standard Part of Speech tagset for the Arabic Language // Journal of King Saud University – Computer and Information Services. – Amsterdam: Elsevier B.V. – 2017. – Т. 29. – Вып. 2. – 562 с. – С. 171 – 178.