Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»
Физтех-школа Радиотехники и Компьютерных Технологий
Кафедра проблем передачи информации и анализа данных

Направление подготовки / специальность: 03.03.01 Прикладные математика и физика
Направленность (профиль) подготовки: Радиотехника и компьютерные технологии

ИССЛЕДОВАНИЕ ПОРЯДКА ПРЕДСКАЗАНИЯ ГРАММЕМ В ЗАДАЧЕ АВТОМАТИЧЕСКОЙ МОРФОЛОГИЧЕСКОЙ РАЗМЕТКИ НА МАТЕРИАЛЕ РУССКОГО ЯЗЫКА

(бакалаврская работа)

Студент:
Щербак Никита Сергеевич

_____
(подпись студента)

Научный руководитель:
Мовсесян Андрей Арсенович,
канд. техн. наук

_____
(подпись научного руководителя)

Консультант *(при наличии)*:
_____
_____
(подпись консультанта)

Москва 2025

Аннотация

Исследование порядка предсказания граммем в задаче автоматической морфологической разметки на материале русского языка

Щербак Никита Сергеевич

Данная работа посвящена задаче автоматической морфологической разметки. Существующие работы показывают, что качество разметки выше, если моделировать граммемы как составные части тега. При этом среди моделей, которые это делают, лучшее качество показала SEQ, которая генерирует граммемы последовательно, используя результат последнего предсказания для генерации следующей граммемы. Другие работы показывают, что в других задачах порядок генерации выходных данных может оказывать существенное влияние. Однако нет работ, которые изучают влияние порядка в контексте морфологической разметки. Данная работа посвящена исследованию того, зависит ли качество морфологической разметки от порядка предсказания граммем. Для этого в ходе работы реализована модель SEQ и проведены эксперименты для разных порядков предсказания, которые были выбраны на основе литературы. Результаты показали, что порядок оказывает влияние как на общее качество разметки, так и на качество предсказания отдельных граммем. Также проведен качественный анализ с целью объяснения этой зависимости. Полученные результаты можно использовать для улучшения качества существующих систем автоматической морфологической разметки.

Содержание

1 Введение

Предметом изучения компьютерной лингвистики являются языки, используемые людьми для общения, которые называют естественными в силу их природы. Задачей компьютерной лингвистики является описание этих языков и построение математических моделей, описывающих их структуру с целью выявления закономерностей и формализации. Одним из разделов лингвистики является морфология, которая изучает слова естественного языка. Задачей морфологии является описание слова как языкового объекта. В рамках морфологии для слов вводят понятия морфологических категорий и соответствующих им наборов значений. Например, в русском, как и во многих других языках, часть речи является морфологической категорией, а «имя существительное», «глагол» и т. д. — ее значениями. Русский язык обязывает выражать в тексте или речи приписанные словам морфологические категории, которые могут зависеть в том числе от контекста. Так, например, имена прилагательные в русском языке всегда имеют категорию числа, то есть каждое прилагательное обязано стоять либо в единственном, либо во множественном числе. В то же время, для глаголов число необязательно: в предложении «Как быть?» глагол «быть» не имеет числа. Морфологическое значение в рамках этой работы называется граммемой. Граммемы, среди прочего, различают формы одного и того же слова и обеспечивают грамматическую связность текста. Тем не менее, есть слова, форма которых не меняется при изменении граммем. Таким, например, является неизменяемое слово «кофе», которое во фразе «эта чашка кофе» стоит в именительном падеже, а во фразе «этой чашкой кофе» — в творительном. Таким образом, граммемы слова зависят от контекста, в котором оно стоит.

Корпусами в контексте компьютерной лингвистики называют собрания текстов на естественных языках. Они используются для исследования языка, для статистического анализа или анализа эволюции языка во времени. Кроме того, они могут использоваться как источники данных для разработки различных языковых ресурсов и моделей в областях компьютерной лингвистики и обработки естественного языка. Аннотированные, или размеченные, корпусы содержат кроме текста дополнительную информацию о нем или его содержимом. В частности, в морфологически аннотированных корпусах каждому слову каждого предложения поставлены в соответствие граммемы. Набор граммем для конкретного слова называется тегом и полностью описывает его грамматические свойства. Из-за большого объема данных, полностью ручная разметка является невыполнимой задачей. В связи с этим разрабатываются автоматические системы.

Под автоматической морфологической разметкой обычно понимают систему, которая каждому слову заданного предложения на естественном языке ставит в соответствие тег. Иными словами, данными, поступающими на вход системе, является текст или собрание текстов, а результатом ее работы — морфологически аннотированный корпус. В свою очередь, морфологически аннотированные корпусы являются источником данных для разработки таких систем с помощью методов обучения с учителем. Поскольку никакая система не гарантирует отсутствие ошибок, автоматически размеченные корпусы текстов обычно проверяются и редактируются лингвистами. Тем не менее, автоматические системы часто позволяют с высокой точностью разметить набор текстов. В основе таких систем могут лежать, например, правиловые подходы, которые при определении тега слова следуют набору строгих инструкций, например [1]. Они часто используются для языков с небольшими объемами корпусов, которых недостаточно для качественной работы нейросети. Однако правиловые системы имеют недостатки. Для их реализации требуется глубокое знание морфологии конкретного языка. В связи с этим активно развиваются именно нейросетевые методы, позволяющие обучать нейросети на небольших объемах данных (см. [2]). В рамках этой работы также рассматривалась нейросетевая разметка.

Сам процесс предсказания тоже может быть разным. Так, система может предсказывать весь тег сразу, не отделяя граммемы одну от другой, или использовать для каждой граммемы отдельный классификатор. Другим вариантом является предсказание граммем последовательно, используя последнюю предсказанную для генерации следующей, что исследовано, например, в работе [3]. В [4] авторы показали, что среди моделей, которые моделируют граммемы как составные части полного тега, лучшую показывает именно та модель, которая генериует граммемы последовательно. Такой вариант является предметом исследования в этой работе. При таком режиме предсказания естественным образом возникает вопрос, будет ли зависеть качество автоматической разметки от порядка предсказания. Однако авторы [4] и других работ это не исследовали. В случае, если зависимость от порядка есть, причины ее могут быть связаны с взаимодействием граммем. Вопрос взаимодействия также представляет интерес и исследуется лингвистами с теоретической точки зрения (см., например, [5]).

Вопрос влияния порядка предсказания элементов в моделях, которые предсказывают выходные данные последовательно, исследовался ранее. В частности, в статье [6] авторы на примерах различных задач продемонстрировали, что порядок действительно оказывает влияние. Тем не менее, влияние порядка предсказания никогда раньше

не исследовалось в контексте морфологической разметки. В качестве модели, на основе которой проводилось исследование, выбрана модель из [4]. Это обусловлено тем, что авторы использовали в качестве данных большое число корпусов разных языков, а значит их модель имеет широкое практическое применение.

Цель этой работы — выяснить, влияет ли порядок, в котором система предсказывает граммемы, на качество морфологической разметки. Для достжения этой цели нужно решить следующие **задачи**:

1. Реализовать модель, предложенную в [4].
2. Добавить в модель возможность выбора порядка генерации.
3. Провести эксперименты.
4. Провести количественный анализ результатов предсказания с использованием разных порядков.
5. Определить причины полученных результатов при помощи качественного анализа.

Апробация: часть результатов данного исследования была представлена на 67-ой Всероссийской научной конференции МФТИ.

2 Постановка задачи морфологической разметки

Для построения систем нейросетевой морфологической разметки применяются методы обучения с учителем. Для этого используются морфологически размеченные корпусы. Существуют разные форматы разметки морфологических корпусов, которые могут отличаться как доступным набором граммем, так и тем, присвоены ли граммемы знакам пунктуации. При решении задачи автоматической морфологической разметки, например, в рамках соответствующего соревнования SIGMORPHON-2019 [7], общепринято рассматривать знаки пунктуации отдельно от слов, приписывая им значение «пунктуация» категории «часть речи». Эта работа не является исключением, поэтому в дальнейшем любой объект, имеющий граммемы, будет называться токеном.

Приведем структуру морфологически размеченного корпуса. Любой такой корпус разделен на предложения. Каждое предложение s состоит из n токенов $\{w_1, w_2, ..., w_n\}$. Тег слова w_i обозначим t_i — он представляет собой последовательность граммем $t_i = \{f_{i1}, f_{i2}, ..., f_{im}\}$. При этом порядок, в котором стоят граммемы, зависит от формата разметки. Итак, корпус — это собрание текстов, которые состоят из предложений, которые состоят из токенов. На Рис. 1 приведен пример того, как может выглядеть предложение в морфологически аннотированном корпусе.

Еще	один	пример	.
наречие положительная ст. сравнения	числительное именительный падеж мужской род	существительное неодушевленное именительный падеж мужской род единственное число	пунктуация

Рис. 1: Пример предложения из морфологически размеченного корпуса

Систему, выполняющую нейросетевую морфологическую разметку, будем называть моделью в соответствии с общепринятой в машинном обучении терминологией. Модель принимает на вход предложение $s = \{w_1, w_2, ..., w_n\}$. Выходом модели является список тегов $\{t_1, t_2, ..., t_n\}$, длина которого равна числу токенов. Каждый тег в списке соответствует одному токену. Порядок, в котором граммемы стоят в тегах, не имеет значения, то есть тег — это множество граммем. Ниже перечислены метрики, использовавшиеся в нашей работе для оценки общего качества моделей.

- Точность — отношение количества слов, у которых правильно определены все грам-

мемы, к общему числу слов в выборке. Формально эту метрику можно записать как

$$accuracy = \frac{\sum_{i=1}^{W} \delta_{\{F_i=Y_i\}}}{W},$$

где W — общее число слов в выборке, за F_i обозначено множество граммем, которые были предсказаны моделью для токена i, за Y_i — истинное множество граммем для него, а $\delta_{\{F_i=Y_i\}}$ — индикатор того, что множества равны.

- Микро-усредненная по граммемам F-мера. Предсказание каждой граммемы рассматривается как бинарная классификация, то есть, например, верно предсказанная граммема для слова, в котором она присутствует в разметке является истинно положительным результатом. Затем стандартным образом рассчитывается микро-усредненная F-мера. Формальная запись этой метрики выглядит следующим образом:

$$F\text{-}мера_{микро} = \frac{\sum_{c=1}^{C}(2 \cdot TP_c)}{\sum_{c=1}^{C}(2 \cdot TP_c + FP_c + FN_c)},$$

где C — это число граммем в выборке, а TP_c, FP_c и FN_c — это число истинно положительных, ложноположительных и ложноотрицательных результатов предсказания для граммемы с номером c.

Помимо качества модели в целом, интерес также представляет качество разметки отдельных граммем. Поэтому для каждой из них также считалась F-мера как для задачи бинарной классификации.

3 Обзор литературы

Для решения задачи морфологической разметки используются разные подходы. Многие из них рассматривают наборы морфологических признаков слов как монолитные теги, предсказывая их целиком с помощью классификатора [8, 9]. Недостатками такого подхода является слишком большое количество возможных значений, многие из которых встречаются слишком редко, а также то, что два тега, отличающиеся только одной граммемой, расцениваются как полностью независимые друг от друга. В связи с этим были разработаны подходы, которые рассматривают теги как множества граммем, которые затем генерируются по одной. Статьи, описывающие такие модели, включают, например, [3]. В ней авторы показали, что рассмотрение граммем как составных частей тега позволяет нейросети находить между ними зависимости, а также генерировать теги, которые не встречались в тренировочной выборке. В [4] авторы провели сравнительный анализ и подтвердили, что моделирование граммем как составных частей полного тега и предсказание каждой из них отдельно действительно улучшает качество разметки. Авторы сравнили между собой три варианта моделей, использующих такой подход: отдельный классификатор для каждой морфологической категории, его модификация, в которой классификаторы используют не только токен, но и предсказанное отдельным классификатором значение категории части речи, а также модель, последовательно генерирующая граммемы. В среднем лучшие результаты показал третий вариант, который был назван авторами SEQ. Другие авторы также использовали последовательную генерацию граммем. Так, в [10] авторы использовали рекуррентную нейронную сеть для решения задачи на материале турецкого языка, являющегося морфологически сложным, и продемонстрировали улучшение качества разметки по сравнению с другими подходами. В [11] авторы показали, что такой способ генерации увеличивает качество разметки тегов, длина которых значительно больше средней.

Последовательная генерация граммем позволяет выбирать порядок, в котором модель будет их предсказывать. Тем не менее, авторы работ, приведенных выше, не проводили исследование разных порядков генерации граммем, используя только один фиксированный порядок, чаще всего определенный корпусом. Однако существуют работы, рассматривающие зависимость качества предсказания других меток, не связанных с морфологией, от порядка в моделях с подобной архитектурой. В [6] авторы показали, что во многих задачах существенное влияние оказывает порядок как входных, так и предсказываемых данных, даже в случаях, когда теоретически качество модели не

должно зависеть от этого порядка. Для демонстрации влияния порядка генерации выходных данных на качество авторы использовали ряд задач, но не углублялись подробно в причины влияния для каждого конкретного случая.

Существуют работы, исследовавшие порядок предсказания в задаче многоклассовой классификации, которую в контексте отдельного слова решаем и мы. В [12] авторы демонстрируют, что точность предсказания метки зависит от того, на какой позиции она находится. При этом зависимость тем больше, чем реже метка встречается в тестовой выборке. Они показывают, что точность предсказания увеличится, если первыми предсказывать метки, на которых модель ошибается чаще. В [13] авторы также отмечают влияние частоты встречаемости класса. Кроме того, они предлагают метод, основанный на обучении с подкреплением, который уменьшает это влияние. Отметим, что в статьях [12, 13] авторы исследуют подходы, предполагающие автоматическое нахождение оптимальных порядков во время обучения.

Лингвистические исследования указывают на то, что граммемы взаимодействуют между собой. Примером взаимодействия может служить функциональная совместимость, то есть способность двух граммем стоять вместе в одном слове или соседних словах. Так, например, употребление повелительного наклонения глагола ограничивает использование в нем времени или лица. Подробнее про этот и другие типы взаимодействия можно прочитать, например, в книгах [5, 14]. Модель может обучиться использовать эту информацию при определении тега. Это подтверждает, что порядок может повлиять на качество разметки при последовательной генерации, хотя теоретически тег слова зависит только от его написания и контекста.

Порядок, в котором граммемы расположены в теге, фиксирован от слова к слову. Иными словами, если теги двух токенов в корпусе совпадают, то для них совпадают и порядки граммем. Порядок граммем и категорий обычно определяется разработчиками корпусов в соответствии с техническими критериями. В корпусах проекта Универсальных зависимостей [15] граммемы указываются после категорий, к которым они относятся, а затем полученные пары сортируются по алфавиту. Глубоко аннотированный корпус СинТагРус [16] использует другой порядок, так что в нем, например, род существительного идет раньше, чем падеж, в отличие от Универсальных зависимостей. В корпусе проекта OpenCorpora [17] используется подобный способ сортировки граммем, но общее число граммем и категорий различается.

4 Описание исследования

4.1 Описание модели

Приведем формальное описание задачи. Как было сказано выше, последовательная генерация граммем дает лучшее качество морфологической разметки среди тех подходов, которые рассматривают граммемы как составные части целого тега. После завершения работы этой модели каждая полученная граммема f_{ij} является элементом списка, который образует тег t_i и соответствует одному токену w_i. Таким образом, моделируется совместное распределение $p(f_{i1}, ..., f_{ik})$. В процессе генерации очередной граммемы в модели SEQ используется последнее предсказание, поэтому вероятность получения тега $t_i = \{f_{i1}, f_{i2}, ..., f_{ik}\}$ при входе w_i можно записать формулой

$$p(t_i|w_i) = \prod_{j=1}^{k} p(f_j|f_0, ..., f_{j-1}, w_i),$$

где k — число сгенерированных граммем в теге. Оно может быть не равно m — числу истинных граммем.

При таком режиме работы возникает предположение, что порядок граммем может влиять на работу модели. Несмотря на то, что для определения полного тега слова достаточно только его и слов контекста, модель может использовать информацию о порядке ввиду недостаточного объема данных или ограниченных вычислительных ресурсов. Для проверки этой гипотезы в этой работе была взята именно модель SEQ из [4], а не более современные модели, использующие такой способ генерации граммем, поскольку изменение порядка скажется на ее работе в большей степени. Целью исследования является не нахождение некого оптимального порядка, который даст абсолютный выигрыш в качестве, а исследование порядков, и оригинальная модель SEQ подходит для этого лучше именно из-за большего количества ошибок, которое позволит в полной мере оценить влияние порядка на качество. Опишем теперь подробнее устройство модели.

Для работы с токенами необходимо перевести их в соответствующие векторные представления. Векторное представление токена w_i будем обозначать h_i. Для получения h_i вне зависимости от конкретной модели используются все слова контекста. В оригинальной модели процедуру перевода токенов в векторные представления выполняет кодер. Обозначим за v_i такой вектор для слова w_i. Перед началом обучения составляется словарь всех токенов, которые встречаются в тренировочной выборке. В качестве исходных векторов слов авторами взяты предобученные векторы библиотеки fastText

([18]), которые в процессе обучения лишь дообучаются. Тем не менее, в тренировочной выборке встречаются токены, которых нет в словаре fastText. Их векторы v_i инициализируются до начала обучения случайно, после чего также добавляются в словарь. Помимо токенов из тренировочной выборки, в словаре также присутствуют специальные токены «PAD», «UNK» и «NUM». «PAD» используется для формирования мини-пакетов предложений в случае, если количество токенов для нескольких предложений из одного мини-пакета не совпадает. К более коротким предложениям «PAD» добавляется до тех пор, пока длина всех предложений не станет равной. Токен неизвестного слова «UNK» используется лишь на этапе тестирования, заменяя токены, которых нет ни в словаре модели, ни в словаре fastText. Векторы токенов «PAD» и «UNK» не изменяются во время обучения. Токен «NUM» заменяет все токены, которые состоят только из цифр. Он используется, поскольку многие числа, встречающиеся в тексте, уникальны для всей выборки, поэтому на этапе тестирования они будут заменяться на токен «UNK» несмотря на то, что контекст у многих таких токенов может быть одинаковым.

Помимо вектора v_i токена и векторов $\{v_1, ..., v_{i-1}, v_{i+1}, ..., v_n\}$ его контекста в кодере для получения представления h_i также используется векторное представление этого токена как последовательности символов, из которых он состоит. Обозначим его c_i. В контексте морфологической разметки символы играют важную роль, поскольку граммемы различают слова на символьном уровне, например, меняя окончание. Кроме того, у пары слов, которые имеют разный контекст, но совпадающие морфы, то есть, например, одинаковые приставки или окончания, векторы v_i могут сильно отличаться, а векторные представления c_i быть близко. Представим, например, что в тестовой выборке встретилось предложение «Смеркалось.», а слова «смеркалось» нет ни в тренировочной выборке, ни в словаре fastText. Тогда его вектор v_1 равен вектору токена «UNK», а единственный токен контекста — это пунктуация, которая в данном случае несодержательна. Тогда единственная информация, которая может использоваться моделью, относится к его символьному представлению. Так, модель может верно определить часть речи «глагол», основываясь на символы «лось», которые присущи другим глаголам в прошедшем времени. В связи с этим на этапе обучения вышеупомянутый токен «UNK» случайным образом заменяет слова, которые в тренировочной выборке встретились лишь один раз. Это стимулирует модель сильнее опираться на символьные представления. Все символы, которые встречаются в тренировочной выборке, так же, как и токены, добавляются в словарь. Векторные представления для них инициализируются случайно и тоже обучаются в процессе. В словаре символов также присутствуют

токен «PAD» для выравнивания слов разной длины и токен «UNK» для замены тех символов из тестовой выборки, которых нет в словаре. Для того, чтобы получить из нескольких векторов символов один вектор c_i для слова, используется двунаправленный слой LSTM [19]. Она принимает на вход последовательность векторных представлений символов, генерируя на выходе один вектор, содержащий контекст всех символов:

$$c_i = BiLSTM_{chars}(\{c_i^1, ..., c_i^l\}),$$

где l — длина слова в символах, а c_i^k — вектор символа на позиции k.

Векторы слов v_i и c_i все еще не содержат контекста предложения. Для его получения они конкатенируются, образуя вектор e_i, а затем используются для получения векторных представлений слов. Конкатенация заключается в простом приписывании вектора c_i к вектору v_i:

$$e_i = [v_i; c_i],$$

то есть длина вектора e_i равна сумме длин v_i и c_i. Наконец, векторы $\{e_1, ..., e_n\}$ поступают на вход другому двунаправленному слою LSTM, выходом которой служит последовательность $\{h_1, ..., h_n\}$. Принцип работы этого слоя LSTM изображен на Рис. 2. На вход каждой ячейке поступает вектор токена e_i и полученное на предыдущем шаге скрытое представление. В качестве начальных скрытых представлений, поступающих в ячейку на первом шаге, выступают нулевые векторы, которые на картинке обозначены нулем.

Часть модели, которая использует векторные представления $\{h_1, ..., h_n\}$ для получения тегов $\{t_1, ..., t_n\}$, называется декодером. Он представляет собой однонаправленный слой LSTM, описание работы которой приведено ниже. Поскольку к моменту декодирования весь контекст токенов содержится в их векторных представлениях, для генерации тега t_i используется только h_i. Для простоты изложения, индекс i в оставшейся части этого раздела опускается.

Для того, чтобы использовать последнюю предсказанную граммему f_{j-1} для получения следующей граммемы f_j, необходимо и граммемам присваивать векторные представления. Для этого, аналогично случаю с токенами и символами, перед началом обучения создается словарь граммем. В него помимо всех граммем, встречающихся в тренировочной выборке, включаются также специальные граммемы: «PAD», используемая на этапе тренировки для дополнения всех тегов мини-пакета до длины мак-

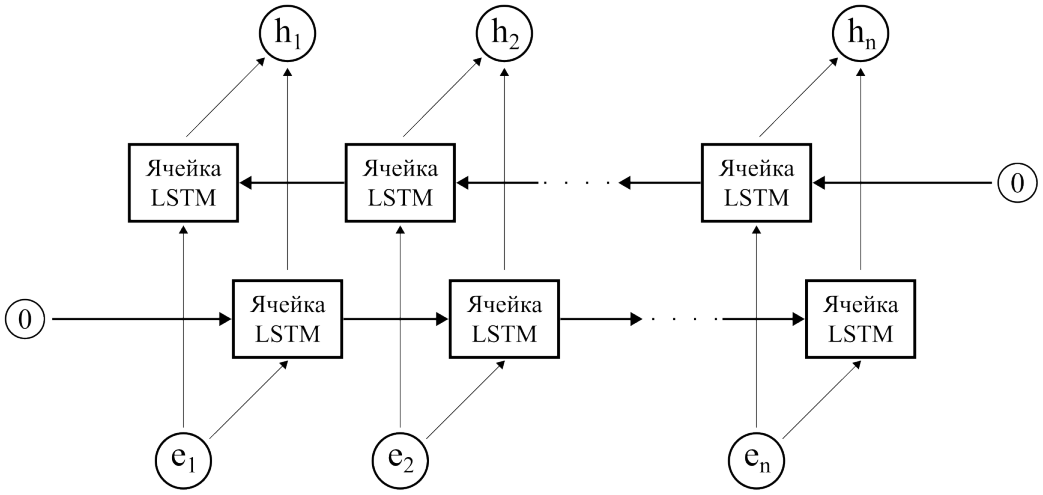

Рис. 2: Получение векторных представлений $\{h_1, ..., h_n\}$

симального тега в нем; «SOS», символизирующая начало последовательности; «EOS», символизирующая конец последовательности. О том, как используются «SOS» и «EOS», будет сказано ниже. Векторы граммем, как и в случае с символами, инициализируются случайным образом.

Как было сказано выше, декодер использует векторное представление h для генерации граммем. При этом после каждого шага, то есть после каждой сгенерированной граммемы, оно изменяется. Это измененное представление и используется для опреления, какая граммема была предсказана на этом шаге. Обозначим за g_j полученное после шага j векторное представление. Для того, чтобы получить из g_j граммему f_j, используется линейный слой, с помощью которого получается вектор x_j длины, равной длине словаря граммем. Получение из него граммемы сводится к взятию индекса максимального значения, который указывает на номер граммемы в словаре. Векторное представление этой граммемы, взятое из словаря, затем поступает на вход следующей ячейке декодера вместе с векторным представлением g_j. Схематически предсказание граммемы на шаге j изображено на Рис. 3

Таким образом, векторы g_j являются по сути скрытыми векторными представлениями модели LSTM.

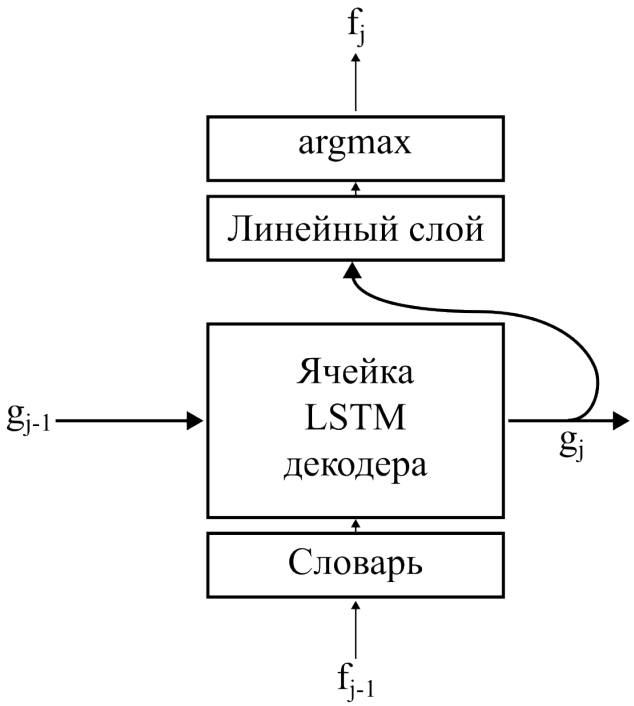

Рис. 3: Устройство ячейки декодера

На вход первой ячейке в качестве векторного представления граммемы используется векторное представление «SOS». При этом в процессе обучения используется форсированное обучение (teacher forcing), то есть вместо последней предсказанной граммемы используется истинная. На этапе обучения к последовательности граммем слова приписывается специальная граммема «EOS». В отличие от граммем «PAD» и «SOS», она учитывается при подсчете значения функции потерь, что стимулирует модель генерировать последовательность правильной длины. Количество шагов генерации следующей граммемы на этапе обучения равно максимальной для всего мини-пакета предложений длине тега, а на этапе тестирования задается гиперпараметром. Токены, которые были сгенерированы декодером после граммемы «EOS» не считаются принадлежащими тегу токена и не учитываются при подсчете функции потерь. Принцип работы декодера как целого на примере предложения изображен на Рис. 4.

В качестве основной функции потерь используется перекрестная энтропия, что означает увеличение ее значения при несоблюдении моделью выбранного порядка предсказания. Тем не менее, итоговый тег, который присваивается токену, не зависит от этого порядка. Иными словами, тег считается предсказанным правильно в том случае, если модель сгенерировала до граммемы «EOS» те и только те граммемы, которые принадлежат токену согласно разметке. На вход она принимает полученные после линейного

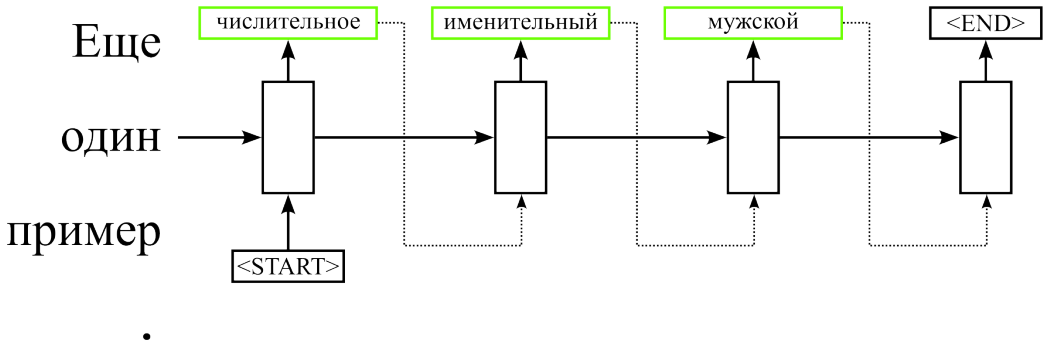

Рис. 4: Архитектура модели SEQ

слоя векторы, которые выступают в роли векторов вероятностей. Для граммемы f_g токена w_i формула подсчета значения функции потерь имеет следующий вид:

$$l_{ij} = -\log \frac{\exp(x_{ij}^{(y_{ij})})}{\sum_{c=1}^{C} \exp(x_{ij}^{(c)})} \cdot \delta_{\{y_{ij} \neq \text{PAD}\}},$$

где y_{ij} — истинная граммема, которая стоит на этом месте, верхний индекс (y_{ij}) равен индексу этой граммемы в словаре, C — количество граммем в словаре, а $\delta_{\{y_{ij} \neq \text{PAD}\}}$ — это индикатор, который используется, чтобы граммема «PAD» не учитывалась при подсчете функции потерь. После того, как посчитаны все такие значения для всех граммем в мини-пакете, считается их среднее, которое является итоговым значением функции потерь этого мини-пакета:

$$\ell(x, y) = \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{l_{ij}}{\sum_{i=1}^{N} \sum_{j=1}^{M} \delta_{\{y_{ij} \neq \text{PAD}\}}},$$

где N — количество токенов в мини-пакете, а M — максимальная по мини-пакету длина тега в граммемах. Отметим, что N равно максимальной по мини-пакету длине предложения в токенах, умноженной на число предложений нем, поскольку более короткие предложения, если есть, дополняются до максимальной длины токеном «PAD», который в качестве граммем также имеет граммемы «PAD».

4.2 Описание корпусов

В качестве данных, на которых обучалась модель, были, как и у авторов модели SEQ, взяты корпусы GSD и SynTagRus версии 2.1 проекта Универсальных зависимостей. Мы также рассмотрели последние на момент исследования версии тех же корпусов

— 2.15. Кроме корпусов из проекта Универсальных зависимостей, представляют интерес корпусы с другими форматами разметки. В связи с этим также был использован корпус проекта OpenCorpora со снятой омонимией версии 0.12. Поскольку в его оригинальной разметке граммемы по большей части не относятся ни к одной морфологической категории, что отличает его от корпусов Универсальных зависимостей, некоторые из таких граммем были сгруппированы в новые категории. Это оставило неизменным количество граммем в словаре и теги слов, изменив лишь количество категорий.

В Таблице 1 приведена статистика тренировочных выборок корпусов. SynTagRus на порядок больше других корпусов по количеству токенов и предложений. Количество уникальных тегов в корпусах отличается не так значительно. Заметим, что меньший по размеру корпус OpenCorpora имеет даже больше уникальных тегов, чем SynTagRus версии 2.1.

Статистика тренировочных выборок корпусов — Таблица 1

	GSD 2.1	SynTagRus 2.1	GSD 2.15	SynTagRus 2.15	OpenCorpora
Число предложений	3850	48814	3850	69630	8534
Число токенов	75964	870034	74900	1204640	56561
Уникальных тегов	693	722	631	1082	1009
Средняя длина тега	3,6	3,5	3,7	3,5	3,3
Максимальная длина тега	10	9	10	9	10

Как было сказано в обзоре литературы, многие теги в корпусе встречаются слишком редко. На Рис. 5 приведено распределение тегов на примере корпуса SynTagRus версии 2.1, которое подтверждает эту несбалансированность. Для других корпусов распределение имеет похожий вид, поэтому здесь не приводится.

Для анализа данных существенную роль будет также играть распределение самих граммем и категорий. В Приложении А приведена визуальная статистика распределения граммем для тренировочных выборок всех пяти упомянутых корпусов. В приложении Б приведена статистика распределения категорий для тренировочных выборок тех же корпусов.

4.3 Рассмотренные порядки предсказания граммем

При выборе порядков для рассмотрения в нашей работе мы опирались на литературу. Изначально мы взяли следующие порядки:

Рис. 5: Распределение тегов, SynTagRus версии 2.1

- соответствующий корпусу, который в дальнейшем будем называть стандартным, поскольку авторы модели SEQ использовали его;
- соответствующий убыванию частоты встречаемости граммем в тренировочной выборке, поскольку в статьях [12, 13] было показано, что точность предсказания класса зависит от частоты. Такой порядок также использовали авторы [11]. Для него введем обозначение «Граммемы↓»;
- обратные им. Порядок, соответствующий возрастанию частоты встречаемости граммем, будет обозначаться «Граммемы↑».

После изучения результатов разметки с этими порядками, мы дополнительно рассмотрели те же порядки, но с частью речи на первой позиции. Такой выбор связан с тем, что часть речи наиболее полно представлена во всех корпусах, поэтому может оказать существенное влияние, если предсказывать её первой. Для таких порядков к названию в начале добавлена пометка «часть речи», например «Часть речи, Граммемы↑». Кроме того, мы добавили порядки, в которых по частоте употребления отсортированы не граммемы, а категории, к которым они относятся. Это сделано для того, чтобы модель выучила их порядок и на этапе предсказания имела большую вероятность предсказать редкую граммему, относящуюся к частой категории, на основе того, что про эту категорию известно из векторного представления слова. Так, например, местный падеж встречается редко, но образует новую форму слова так же, как это делают другие падежи — изменением окончания. Это дает основания предсказывать его на том же этапе, на котором предсказываются другие падежи. Для порядков, использующих категории для сортировки по частоте, в названии используется слово «Категории».

За выбор порядка в нашей модели отвечает отдельный гиперпараметр. Он определяет то, в каком порядке граммемы подаются в декодер на этапе обучения, а также за то, какую позицию занимает та или иная граммема токена при подсчете функции потерь.

4.4 Функция потерь, не зависящая от порядка

Поскольку перебор всех порядков на практике неосуществим, мы добавили ещё один режим обучения, при котором функция потерь зависит только от набора предсказанных граммем, но не от порядка непосредственно. Она была описана в [20] и представляет собой модификацию перекрестной энтропии, в связи с чем названа независящей от порядка перекрестной энтропией (order-agnostic cross entropy, OAXE). OAXE учитывает все возможные перестановки выходных данных и находит среди них ту, что даст минимальное значение стандартной перекрестной энтропии. Таким образом, модель постепенно автоматически выбирает тот порядок, который является оптимальным, и использует его. Для этого она использует векторы $\{x_{i1}, ..., x_{ik}\}$, которые были введены выше. k — это фиксированное число, гиперпараметр модели, который выбирается большим, чем максимальная длина тега в корпусе. Из них берутся только первые m векторов (здесь m, как и раньше, равно числу истинных граммем этого токена). Затем из векторов $\{x_{i1}, ..., x_{im}\}$ составляется матрица размера $m \times C$. Для того, чтобы найти перестановку, из нее берутся только те столбцы, которые соответствуют истинным граммемам. Иными словами, берутся коэффициенты, которые модель получила для m истинных граммем за m первых шагов генерации. Таким образом получается квадратная матрица, которая используется как матрица стоимости в задаче о назначениях (более подробно про эту задачу см., например, [21]). Далее находится оптимальная перестановка граммем токена, которая в итоге используется для подсчета стандартной перекрестной энтропии, которая описана выше.

4.5 Проделанная работа

В этом разделе частично описана работа, проделанная для достижения целей исследования.

В ходе работы была воспроизведена без изменений модель SEQ из [4]. Оригинальная модель написана на устаревшей к настоящему моменту версии библиотеки TensorFlow [22], поэтому она была переписана на актуальной версии PyTorch [23]. При этом в коде авторов была обнаружена и исправлена ошибка, которая приводила к обу-

чению токена «PAD». В остальном логика работы и выбранные гиперпараметры полностью совпадают с тем, что предложили авторы.

Для выбора порядка предсказания в модель был добавлен соответствующий гиперпараметр. Кроме него, была также реализована независящая от порядка функция потерь (OAXE). За выбор между ней и стандартной перекрестной энтропией отвечает отдельный гиперпараметр. Для исследования влияния порядков были также реализованы функции подсчета метрик, которые описаны выше.

К технической части этой работы относится и работа, проделанная над корпусом OpenCorpora. Файлы с оригинальной разметкой не подходят для использования в модели, так как в них граммемы указаны без категорий, поэтому из них были сгенерированы новые файлы, которые это учитывают. Кроме того, как было упомянуто выше, часть граммем, которые в корпусе не относятся ни к одной категории, были объединены вместе. Так, например, граммемы, относящиеся к окончаниям слов («V-oy» для слов на *-ой*, «V-ey» для слов на *-ей*, и т. п.), стали относится к одной категории «Ending» (см. также Приложение Б).

4.6 Эксперименты

Перейдем к описанию экспериментов. Модель была обучена на каждом корпусе с каждым из порядков 5 раз. После этого все модели использовались для подсчета метрик на тестовой выборке каждого корпуса. Одной из метрик, как было сказано, является точность. Она является у авторов оригинальной модели основной метрикой, что позволяет сравнить наши результаты. Кроме того, если автоматическая морфологическая разметка используется для создания аннотированных корпусов, точность представляет интерес, так как показывает, какой процент слов лингвисту не придется проверять вручную.

Кроме точности, для всех порядков также была посчитана микро-усредненная F-мера. Она используется, поскольку показывает, насколько хорошо модель в среднем предсказывает граммемы, то есть является мерой качества самого декодера.

После получения значений этих метрик для каждой из пяти моделей на всех корпусах и порядках, были посчитаны их средние значения и стандартные отклонения, которые приведены в разделе с результатами. Они также использовались для того, чтобы проверить статистическую значимость различий в качестве. Для этого был использован критерий Стьюдента, выборками служили 5 значений конкретной метрики. На каждом из корпусов изначально все порядки сравнивались со стандартным. После

этого для каждой из метрик был выбран худший и лучший по ее среднему значений порядки, и статистическое различие таким же образом было найдено для них. В случаях нескольких худших или лучших порядков выбирался тот, стандартное отклонение метрики на котором меньше. Эти пары порядков затем использовались для качественного анализа. При выборе не учитывалась модель, использующая OAXE, поскольку она не гарантирует, что взаимное расположение граммем не меняется от токена к токену, что затрудняет анализ.

Кроме метрик, показывающих общее качество разметки, как было упомянуто в постановке задачи, для каждой граммемы также была посчитана F-мера. Она может указать на отличия моделей, для которых значения точности или микро-усредненной F-меры близки. Это будет служить свидетельством того, что порядок может влиять на одни группы граммем отрицательно, тогда как на другие — положительно. Для порядков, выбранных вышеописанным образом, и для всех граммем были найдены все токены из тестовой выборки, для каждого из которых использование одного порядка привело к ошибке в предсказании граммемы, а другого — к правильному предсказанию. Однако использование пяти моделей для каждого из поряков означает, что в случае предсказания ими для конкретного токена разных тегов неясно, какой именно тег использовать для проверки на ошибки. В связи с этим было выбрано три варианта:

- граммема считается предсказанной конкрентным порядком, если среди пяти запусков модели для этого порядка она была сгенерирована чаще всего;
- граммема считается предсказанной конкрентным порядком, если она присутствует во всех пяти тегах;
- граммема считается предсказанной конкрентным порядком, если она была сгенерирована той моделью, которая среди других моделей для этого порядка дает на ней лучшее значение F-меры.

Кроме того, для этих же пар порядков были найдены все граммемы со статистически значимым отличием в F-мере. Они также приведены в разделе, посвященном результатам.

5 Результаты

Ниже приведены таблицы со средними значениями и стандартными отклонениями метрик. Полужирным отмечены лучшие значения. Для того, чтобы количественно определить влияние порядка, стандартный порядок сравнивался с остальными при помощи критерия Стьюдента. Значения тех порядков, отличия в качестве на которых от стандартного оказались статистически значимыми на уровне 0,05, в таблицах подчеркнуты.

Таблицы 2 и 3 содержат значения точности. Помимо этого, в Таблице 2 приведены результаты авторов модели SEQ. Как было сказано в предыдущем разделе, наше качество выше авторского из-за ошибки в коде авторов, которая приводила к обучению вектоного представления токена «PAD».

Точность на GSD и SynTagRus версии 2.1 — Таблица 2

Порядок	GSD	SynTagRus
Стандартный	**91,29 ± 0,12**	96,78 ± 0,08
Обратный	91,18 ± 0,09	96,83 ± 0,04
«Часть речи, обратный»	91,19 ± 0,07	96,80 ± 0,05
«Граммемы↓»	91,01 ± 0,12	96,84 ± 0,05
«Часть речи, граммемы↓»	91,19 ± 0,14	96,78 ± 0,08
«Категории↓»	91,14 ± 0,07	96,82 ± 0,09
«Граммемы↑»	91,12 ± 0,07	96,78 ± 0,08
«Часть речи, граммемы↑»	91,15 ± 0,11	**96,87 ± 0,08**
«Категории↑»	91,23 ± 0,09	96,77 ± 0,05
«Часть речи, категории↑»	91,18 ± 0,13	96,80 ± 0,08
OAXE	90,96 ± 0,11	96,71 ± 0,05
Результаты авторов [4]	91,05 ± 0,18	96,67

Для GSD лучшим порядком оказался стандартный. Это может быть связано с тем, что авторы модели SEQ подбирали гиперпараметры в том числе на этом корпусе. Поскольку GSD относительно небольшой, на нем подбор гиперпараметров может оказывать более существенное влияние, чем порядок. В то же время, на большем корпусе SynTagRus лучшее качество показала модель с другим порядком. Несмотря на то, что на этом корпусе нет ни одного порядка, который дал статистически значимое отличие,

Точность на GSD и SynTagRus версии 2.15 и на OpenCorpora — Таблица 3

Порядок	GSD	SynTagRus	OpenCorpora
Стандартный	**92,01 ± 0,13**	93,08 ± 0,07	47,83 ± 0,15
Обратный	91,92 ± 0,17	93,12 ± 0,09	47,61 ± 0,10
«Часть речи, обратный»	91,94 ± 0,11	93,12 ± 0,12	47,79 ± 0,19
«Граммемы↓»	91,88 ± 0,09	**93,16 ± 0,05**	47,78 ± 0,20
«Часть речи, граммемы↓»	91,75 ± 0,19	93,07 ± 0,08	47,82 ± 0,24
«Категории↓»	91,91 ± 0,13	93,15 ± 0,13	47,73 ± 0,27
«Граммемы↑»	91,91 ± 0,05	93,10 ± 0,04	47,53 ± 0,10
«Часть речи, граммемы↑»	91,93 ± 0,12	93,12 ± 0,09	47,69 ± 0,23
«Категории↑»	91,99 ± 0,13	93,05 ± 0,07	47,71 ± 0,20
«Часть речи, категории↑»	91,95 ± 0,14	93,06 ± 0,08	47,73 ± 0,12
OAXE	91,64 ± 0,12	88,40 ± 0,04	**47,90 ± 0,20**

порядки могли по-разному повлиять на отдельные граммемы, что оказало небольшое влияние на общую метрику.

Таблицы 4 и 5 содержат значения микро-усредненной F-меры.

Микро-усредненная F-мера на GSD и SynTagRus версии 2.1 — Таблица 4

Порядок	GSD	SynTagRus
Стандартный	96,23 ± 0,08	98,54 ± 0,03
Обратный	**96,24 ± 0,06**	98,54 ± 0,01
«Часть речи, обратный»	96,18 ± 0,04	98,54 ± 0,02
«Граммемы↓»	96,14 ± 0,06	98,54 ± 0,02
«Часть речи, граммемы↓»	96,20 ± 0,03	98,53 ± 0,03
«Категории↓»	96,16 ± 0,05	98,56 ± 0,04
«Граммемы↑»	96,12 ± 0,04	98,53 ± 0,03
«Часть речи, граммемы↑»	96,18 ± 0,07	**98,57 ± 0,02**
«Категории↑»	96,19 ± 0,08	98,52 ± 0,02
«Часть речи категории↑»	96,17 ± 0,05	98,54 ± 0,03
OAXE	96,23 ± 0,05	98,56 ± 0,02

Относительное по качеству предсказания взаимное расположение порядков меня-

Микро-усредненная F-мера на GSD и SynTagRus версии 2.15 и на OpenCorpora — Таблица 5

Порядок	GSD	SynTagRus	OpenCorpora
Стандартный	96,38 ± 0,07	97,44 ± 0,04	97,60 ± 0,08
Обратный	96,33 ± 0,05	97,46 ± 0,04	97,49 ± 0,12
«Часть речи, обратный»	96,32 ± 0,02	**97,47 ± 0,06**	97,65 ± 0,12
«Граммемы↓»	96,20 ± 0,06	**97,47 ± 0,03**	97,47 ± 0,11
«Часть речи, граммемы↓»	96,21 ± 0,05	97,44 ± 0,02	97,52 ± 0,09
«Категории↓»	96,33 ± 0,06	**97,47 ± 0,06**	97,51 ± 0,10
«Граммемы↑»	96,34 ± 0,06	97,46 ± 0,03	97,49 ± 0,04
«Часть речи, граммемы↑»	96,31 ± 0,04	97,46 ± 0,02	97,54 ± 0,08
«Категории↑»	96,36 ± 0,03	97,43 ± 0,03	97,58 ± 0,07
«Часть речи, категории↑»	96,32 ± 0,06	97,43 ± 0,05	97,54 ± 0,10
OAXE	**96,41 ± 0,04**	97,29 ± 0,01	**97,70 ± 0,11**

ется в зависимости от используемой метрики. Так, например, на GSD версии 2.1 обратный порядок показал одно из худших значений точности, но лучшее значение микро-усредненной F-меры. То же самое относится и к модели, использующей OAXE на корпусе GSD версии 2.15. Это может быть свидетельством того, что для разных групп граммем лучшими являются разные порядки.

Качество для модели, использующей OAXE, для корпуса SynTagRus версии 2.15 указано для модели, обученной на корпусе SynTagRus версии 2.1. Результат показывает существенное различие в разметки двух версий корпусов, что объясняет и различие в качестве их разметки.

Как было сказано в разделе 4, для каждой из метрик была выбрана пара порядков, один из которых показал лучшее значение, а другой — худшее. Различия для порядков затем были проверены при помощи критерия Стьюдента. В Таблицах 6 и 7 приведены p-значения, жирным выделены значения меньше 0,05. Первым в таблицах идет порядок, показавший лучшее качество.

Для пар порядков, выбранных исходя из точности, граммемы, которые были отобраны согласно описанию экспериментов, приведены в Таблицах 8—9. Пары порядков, выбранные на основе F-меры, для обеих версий SynTagRus совпадают, для остальных корпусов граммемы для таких пар порядков приведены в Таблице 10. Граммемы в

p-значения для различий между худшим и лучшим порядками по точности — Таблица 6

Корпус	Порядок	p-значение
GSD версии 2.1	Стандартный	**0,001**
	«Граммемы↓»	
SynTagRus версии 2.1	«Часть речи, граммемы↑»	0,052
	«Категории↑»	
GSD версии 2.15	Стандартный	0,052
	«Часть речи, граммемы↓»	
SynTagRus версии 2.15	«Граммемы↓»	**0,033**
	«Категории↑»	
OpenCorpora	Стандартный	**0,011**
	«Граммемы↑»	

p-значения для различий между худшим и лучшим порядками по F-мере — Таблица 7

Корпус	Порядок	p-значение
GSD версии 2.1	Обратный	**0,012**
	«Граммемы↑»	
SynTagRus версии 2.1	«Часть речи, граммемы↑»	**0,020**
	«Категории↑»	
GSD версии 2.15	Стандартный	**0,004**
	«Граммемы↓»	
SynTagRus версии 2.15	«Граммемы↓»	0,063
	«Категории↑»	
OpenCorpora	«Часть речи, обратный»	**0,033**
	«Граммемы↑»	

таблицах сгруппированы в соотвествии с тем, какой порядок выигрывает в качестве. Напротив каждой граммемы указано суммарное по трем вариантам количество токенов, которые были отобраны по этой граммеме вышеописанным способом. В таблицах указаны только те граммемы, соответствующее которым количество ошибок не меньше 10.

Граммемы со статистически значимым отличием для порядков, выбранных по точности на GSD и SynTagRus версии 2.1 — Таблица 8

	Порядки	Граммема	Количество ошибок
GSD версии 2.1	Стандартный	POS=NOUN	80
		POS=PROPN	37
		POS=VERB	26
		Aspect=Perf	31
		Gender=Fem	79
SynTagRus версии 2.1	«Часть речи, граммемы↑»	POS=SCONJ	49
		POS=CCONJ	34
		Mood=Imp	14
		Foreign=Yes	45
		Case=Nom	416
		Case=Gen	237
	«Категории↑»	Case=Loc	65
		Animacy=Anim	211

Для каждой пары порядков есть по крайней мере несколько граммем, на которых отличия F-меры существенны. При этом количество токенов, на которых тот порядок из пары, который является в целом лучшим, сгенерировал граммему верно, достигает нескольких сотен, что означает достаточно репрезентативную выборку, которая подходит для качественного анализа.

Как показал анализ выше, основной вклад в различие качества моделей для каждого корпуса дает небольшой набор граммем. При этом список таких граммем зачастую разный для разных корпусов. С одной стороны, это объясняется различием в объемах корпусов и в их жанровых составах. С другой стороны, корпусы в составе проекта Универсальных зависимостей при переходе с версии 2.1 на версию 2.15 не только претерпели текстовые изменения, но и сам формат разметки существенно изменился: изменилась разметка ряда слов и добавились новые граммемы.

Качественный анализ показывает, что различия в качестве обусловлены порядком предсказания граммем. Рассмотрим несколько примеров из разных корпусов. Здесь и далее если пример частично сопровожден разметкой обеих моделей, то разметка первой упоминаемой модели всегда приводится сверху, а второй — снизу.

Граммемы со статистически значимым отличием для порядков, выбранных по точности для GSD и SynTagRus версии 2.15 и OpenCorpora— Таблица 9

	Порядки	Граммема	Количество ошибок
GSD версии 2.15	Стандартный	POS=ADP	12
		Aspect=Perf	32
		Case=Gen	109
		Case=Nom	107
		Gender=Fem	71
SynTagRus версии 2.15	«Граммемы↓»	POS=NOUN	231
		POS=DET	169
		Reflex=Yes	79
		POS=X	11
	«Категории↑»	Person=2	14
OpenCorpora	Стандартный	POS=NUMB	66
		Aspect=Perf	19
		Transitivity=Tran	26
	«Граммемы↑»	Animacy=Anim	27

Пример (1) взят из корпуса SynTagRus версии 2.1:

(1) Отличие в том, что хозяйственные объекты, которые можно
POS=ADJ|Case=Nom
Animacy=Inan|Case=Acc|POS=ADJ
наблюдать при шанхайском аэропорте, не кончаются до самого горизонта.

Здесь модель с порядком «часть речи, граммемы↑», не допустила ошибок в разметке прилагательного (POS=ADJ) *хозяйственные*, тогда как модель «категории↑», вместо именительного падежа (Case=Nom) предсказала винительный (Case=Acc).

В этом примере подлежащим является существительное *объекты*, а сказуемым, отделенным от подлежащего придаточным, является глагол *кончаются*. Прилагательное *хозяйственные* стоит в именительном падеже, поскольку должно быть согласовано с подлежащим, которое им управляет.

Глагольная группа *можно наблюдать* здесь играет роль аттрактора, то есть словосочетания, которое может помешать установлению корректной связи между подле-

Граммемы со статистически значимым отличием для порядков, выбранных по F-мере для GSD обеих версий и OpenCorpora — Таблица 10

	Порядки	Граммема	Количество ошибок
GSD версии 2.1	Обратный	VerbForm=Part	29
		Aspect=Perf	30
		Aspect=Imp	33
		Animacy=Inan	143
		POS=VERB	33
		POS=ADJ	45
	«Граммемы↑»	POS=DET	15
GSD версии 2.15	Стандартный	POS=ADJ	58
		POS=NUM	17
		POS=VERB	27
		Aspect=Perf	33
		Case=Gen	108
		Case=Nom	122
		Degree=Pos	64
		Tense=Past	15
		VerbForm=Part	19
		Voice=Pass	15
OpenCorpora	«Часть речи, обратный»	POS=NUMB	72
		Transitivity=Tran	26
		Tense=Past	10
		Gender=Masc	46

жащим и сказуемым. Ср. с примером (2), где прилагательное стоит уже в винительном падеже:

(2) Отличие в том, что хозяйственные объекты можно наблюдать.
Case=Acc

Прилагательные в винительном падеже отличается от остальных тем, что имеют одушевленность (ср. *видеть красивую майку* и *видеть красивого зверька*). В примере (1) из-за аттрактора с точки зрения модели может возникнуть неоднозначность: либо считать, что прилагательное стоит в именительном падеже, либо считать, что это неоду-

шевленное прилагательное в винительном падеже. Неуверенность модели с порядком «категории↑» привела к тому, что была предсказана неодушевленность (Animacy=Inan), которая предсказывается перед падежом, так как категория одушевленности — более редкая, чем категория падежа. Неодушевленность прилагательного автоматически требует винительного падежа, что и привело к ошибке. В то же время, сама граммема неодушевленности — более частотная, чем граммема именительного или винительного падежа, поэтому модель с порядком «часть речи, граммемы↑» имела выбор и предсказала именительный падеж, что, в свою очередь, исключает возможность предсказания значения категории одушевленности.

Похожая ситуация наблюдается при выборе части речи в примере (3) из корпуса SynTagRus версии 2.15:

(3) Специалистов нужно 25,5 тыс., служащих (POS=NOUN / Tense=Pres|POS=VERB) 11 тыс., а руководителей всего 2,5 тыс.

Здесь, как и в примере (1), нет неоднозначности: слово *служащих* является существительным (POS=NOUN). Однако выбор между существительным и глаголом (POS=VERB) в форме причастия понятен. Ср. с примером (4):

(4) Специалистов нужно 25,5 тыс., служащих (POS=VERB|Tense=Pres) офицеров нужно 11 тыс., а руководителей всего 2,5 тыс.

Граммема существительного встречается в корпусе чаще любой другой граммемы, присущей глаголу, поэтому в модели с порядком «граммемы↓» она была предсказана первой. А в модели с порядком «категории↑», в силу ее неуверенности, первой была предсказана граммема настоящего времени (Tense=Pres) как граммема одной из самых редких категорий глагола, что исключает возможность предсказания существительного как части речи.

В описанных двух примерах корректная граммема предсказывается моделью, качество предсказаний которой в среднем (с точки зрения F-меры) выше. Однако, как было показано выше (см. Таблицы 8—10), высокое качество предсказаний в среднем не обязательно означает высокое качество предсказаний для каждой отдельной граммемы, поскольку некоторый порядок предсказания может быть эффективным для одной груп-

пы граммем, и неэффективным — для другой. Так, например, происходит в примере (5) из корпуса OpenCorpora:

(5) Подсудимые в Храме не выступали.
Case=Nomn|POS=ADJF
Animacy=Anim|Case=Nomn|POS=NOUN

Для этого корпуса модель с обратным порядком в среднем лучше, чем модель с порядком предсказания «граммемы↑», но в данном примере именно вторая модель корректно предсказала часть речи слова *подсудимые* как существительное, а не прилагательное (POS=ADJF). В качестве прилагательного это слово не используется, так как слово *подсудимый* в современном русском языке субстантивировалось, то есть стало обозначать конкретного человека, однако выбор между прилагательным и существительным понятен. Аналогично примеру (1), предсказание второй моделью одушевленности (Animacy=Anim) перед падежом (Case=Nomn) исключило возможность выбора прилагательного (так как прилагательные в именительном падеже не имеют одушевленности), тогда как первая модель предсказала первым падеж, что привело к неоднозначности того, что предсказывать следующим: одушевленность или сразу часть речи (прилагательное).

Рассмотреные выше примеры связаны с возможностью сочетаемости тех или иных граммем. Однако есть и более сложные случаи. Таким является пример (6) из корпуса GSD версии 2.1:

(6) Самые первые часы, которые кто-либо приспособил для ношения на
Tense=Past|Aspect=Perf
Aspect=Imp
руке, неизвестны.

Здесь модель с обратным порядком предсказания граммем корректно выбрала совершенный вид (Aspect=Perf) глагола *приспособил*, тогда как модель с порядком «граммемы↑» ошиблась, выбрав несовершенный вид (Aspect=Imp). Выбор в пользу совершенного вида здесь однозначный, однако неуверенность модели понятна.

Действительно, у модели есть два источника информации о слове *приспособил*. Во-первых, это векторное представление, полученное из модели fastText. Однако семантически слова *приспособил* и *приспосабливал* близки, поэтому различие в их векторных представлениях может быть незначительным, а контекст не устраняет ноднозначность (оба варианта вполне употребимы в контексте рассматриваемого примера). Во-вторых,

это символьное представление слова. Однако определить совершенный вид может быть затруднительно как по приставке, так и по суффиксу (ср. *приспособил* и *приносил*).

При этом первая модель корректно предсказала вид, поскольку в обратном порядке время стоит раньше, чем вид, тогда как в порядке «граммемы↑» вид предсказывается в самом конце (перед частью речи). Знание о том, что глагол стоит в прошедшем времени (Tense=Past), позволяет выбрать совершенный вид, поскольку из лингвистики известно, что прошедшее время является естественным контекстом для глаголов совершенного вида (подробнее см., например, [24]).

Для корпуса GSD версии 2.1 модель на основе OAXE, которая выбирает порядок предсказания граммем в процессе обучения, не показала лучшего качества. В то же время ручная проверка разметки этой моделью примера (6) показала, что и она первой выбирает значение категории времени. Аналогичная ситуация наблюдается во всех примерах, кроме примера (5): в нем модель предсказала падеж первым, но при этом часть речи определила некорректно только в одном запуске из пяти.

6 Заключение

Целью данной работы было выяснить, влияет ли порядок предсказания граммем в задаче автоматической морфологической разметки на качество. Для этого в ходе работы была реализована модель, которая выполняет процедуру последовательной генерации граммем. В нее добавлена возможность выбора порядка генерации граммем. Исследуемые порядки были выбраны исходя из литературы и первичных экспериментов. Кроме того, был добавлен режим работы модели, при котором используется функция потерь, не зависящая от порядка генерации граммем. Для исследования статистической значимости изменения порядка, использующая каждый порядок модель была обучена 5 раз, после чего полученные метрики качества использовались для количественного анализа, а результаты генерации — для качественного исследования полученных результатов.

Результаты показали, что порядок предсказания действительно влияет на качество морфологической разметки при последовательном предсказании, и это различие статистически значимо. При этом результаты показали, что самые значительные изменения в качестве предсказания наблюдаются на небольшом наборе граммем.

Как показал качественный анализ, причинами изменения являются зависимости между граммемами. Эти зависимости используются моделью как дополнительная информация, что влияет на качество. Модель на основе OAXE, которая выбирает порядок предсказания в процессе обучения, для отобранных нами примеров выбирает такие порядки, которые согласуются с обнаруженными зависимостями.

Полученные результаты демонстрируют, что порядок, в котором граммемы расположены в морфологически аннтированных корпусах, может влиять на результаты работы моделей, которые их используют. В связи с этим есть как минимум два направления для продолжения исследований. Первое направление связано с определением оптимального порядка. Полный перебор всех порядков невозможен, поэтому нужно рассматривать методы автоматического определения порядка. Метод, использующий OAXE, показал свою эффективность, но имеет ряд недостатков и не смог во всех случаях показать наилучшее качество, поэтому необходимо рассматривать другие подходы. Во-вторых, это проверка сделанных выводов на других языках.

Список литературы

[1] *Yoshinaga, N.* Back to Patterns: Efficient Japanese Morphological Analysis with Feature-Sequence Trie / N. Yoshinaga // *arXiv preprint arXiv:2305.19045*. — 2023.

[2] Joint Learning Model for Low-Resource Agglutinative Language Morphological Tagging / Gulinigeer Abudouwaili, Kahaerjiang Abiderexiti, Nian Yi, Aishan Wumaier // Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology / Ed. by Garrett Nicolai, Eleanor Chodroff, Frederic Mailhot, Çağrı Çöltekin. — Toronto, Canada: Association for Computational Linguistics, 2023. — Pp. 27–37. https://aclanthology.org/2023.sigmorphon-1.4/.

[3] *Malaviya, Chaitanya.* Neural factor graph models for cross-lingual morphological tagging / Chaitanya Malaviya, Matthew R Gormley, Graham Neubig // *arXiv preprint arXiv:1805.04570*. — 2018.

[4] *Tkachenko A., Sirts K.* Modeling Composite Labels for Neural Morphological Tagging / Sirts K. Tkachenko A. // Proceedings of the 22nd Conference on Computational Natural Language Learning. — 2018.

[5] Иерархия и взаимодействие грамматических категорий глагола / Ed. by Мальчуков А. Храковский В. — Москва: Институт лингвистических исследований РАН, 2020.

[6] *Vinyals O. Bengio S., Kudlur M.* Order matters: Sequence to sequence for sets. / Kudlur M. Vinyals O., Bengio S. // arXiv preprint. — 2015.

[7] The SIGMORPHON 2019 Shared Task: Morphological Analysis in Context and Cross-Lingual Transfer for Inflection / Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu et al. // Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology / Ed. by Garrett Nicolai, Ryan Cotterell. — Florence, Italy: Association for Computational Linguistics, 2019. — Pp. 229–244. https://aclanthology.org/W19-4226/.

[8] *Müller T., Schütze H.* Robust morphological tagging with word representations / Schütze H. Müller T. // Proceedings of the 2015 Conference of the North

American Chapter of the Association for Computational Linguistics: Human Language Technologies. — 2015.

[9] *Heigold, Georg.* An extensive empirical evaluation of character-based morphological tagging for 14 languages / Georg Heigold, Guenter Neumann, Josef van Genabith // Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. — 2017. — Pp. 505–513.

[10] *Akyürek, Ekin.* Morphological analysis using a sequence decoder / Ekin Akyürek, Erenay Dayanık, Deniz Yuret // *Transactions of the Association for Computational Linguistics.* — 2019. — Vol. 7. — Pp. 567–579.

[11] *Oh, Byung-Doh.* THOMAS: The hegemonic OSU morphological analyzer using seq2seq / Byung-Doh Oh, Pranav Maneriker, Nanjiang Jiang // Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology. — 2019. — Pp. 80–86.

[12] Enhancing Multi-Label Classification via Dynamic Label-Order Learning / Jiangnan Li, Yice Zhang, Shiwei Chen, Ruifeng Xu // *Proceedings of the AAAI Conference on Artificial Intelligence.* — 2024. — Mar. — Vol. 38, no. 17. — Pp. 18527–18535. https://ojs.aaai.org/index.php/AAAI/article/view/29814.

[13] A Deep Reinforced Sequence-to-Set Model for Multi-Label Classification / Pengcheng Yang, Fuli Luo, Shuming Ma et al. // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics / Ed. by Anna Korhonen, David Traum, Lluís Màrquez. — Florence, Italy: Association for Computational Linguistics, 2019. — Pp. 5252–5258. https://aclanthology.org/P19-1518/.

[14] *Мальчуков, Андрей Львович.* Наклонение во взаимодействии с другими категориями: опыт типологического обзора / Андрей Львович Мальчуков, Виктор Самуилович Храковский // *Вопросы языкознания.* — 2015. — no. 6. — Pp. 9–32.

[15] Universal Dependencies / Joakim Nivre, Daniel Zeman, Filip Ginter, Francis Tyers // Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts / Ed. by Alexandre Klementiev, Lucia Specia. — Valencia, Spain: Association for Computational Linguistics, 2017. https://aclanthology.org/E17-5001/.

[16] *Богуславский, И. М.* Современное состояние корпуса СинТагРус / И. М. Богуславский et al. // *Труды Института русского языка им. В. В. Виноградова.* — 2024. — no. 4 (42). — Pp. 141–169.

[17] *Грановский Д. Бочаров В., Бичинева С.* Открытый корпус: принципы работы и перспективы / Бичинева С. Грановский Д., Бочаров В. // Труды научного семинара XIII Всероссийской объединенной конференции «Интернет и современное общество». — 2010.

[18] Enriching word vectors with subword information / Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov // *Transactions of the association for computational linguistics.* — 2017. — Vol. 5. — Pp. 135–146.

[19] *Hochreiter, Sepp.* Long short-term memory / Sepp Hochreiter, Jürgen Schmidhuber // *Neural computation.* — 1997. — Vol. 9, no. 8. — Pp. 1735–1780.

[20] *Du M. Tu Z., Jiang J.* Order-Agnostic Cross Entropy for Non-Autoregressive Machine Translation / Jiang J. Du M., Tu Z. // International conference on machine learning. — 2021.

[21] *Burkard, Rainer E.* Linear assignment problems and extensions / Rainer E Burkard, Eranda Cela // Handbook of combinatorial optimization: Supplement volume A. — Springer, 1999. — Pp. 75–149.

[22] *Abadi, Martín.* TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. — 2015. — Software available from tensorflow.org. https://www.tensorflow.org/.

[23] PyTorch: An Imperative Style, High-Performance Deep Learning Library / Adam Paszke, Sam Gross, Francisco Massa et al. // Advances in Neural Information Processing Systems / Ed. by H. Wallach, H. Larochelle, A. Beygelzimer et al. — Vol. 32. — Curran Associates, Inc., 2019. https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

[24] *Падучева, Е. В.* О семантическом инварианте видового значения глагола в русском языке / Е. В. Падучева // *Русский язык в научном освещении.* — 2004. — Т. 2, № 8. — С. 5–16.

Приложение А

Статистика распределения граммем по корпусам

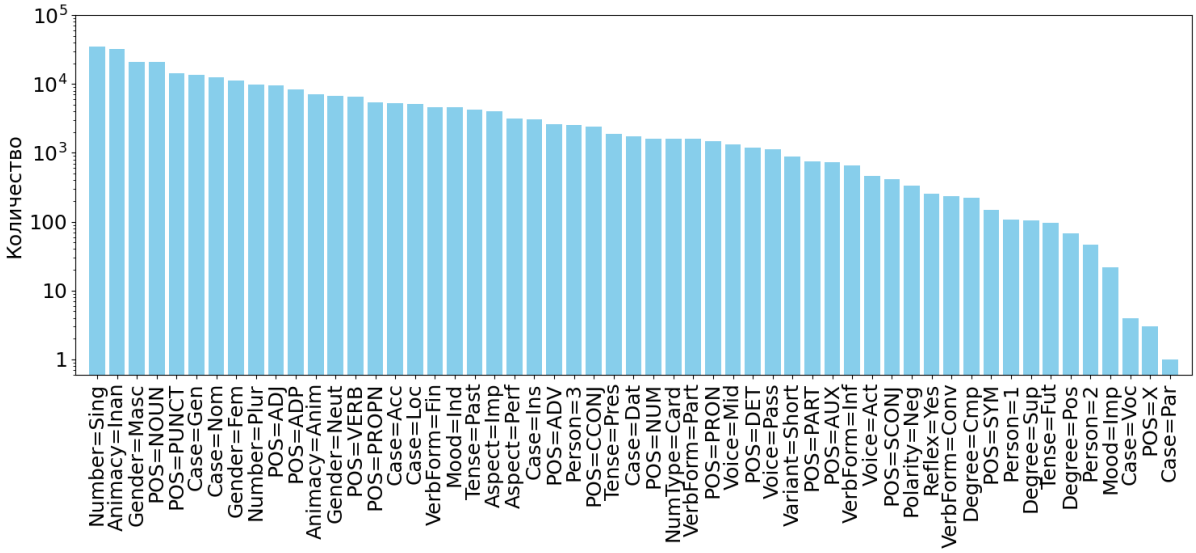

Рис. 6: Распределение граммем, GSD версии 2.1

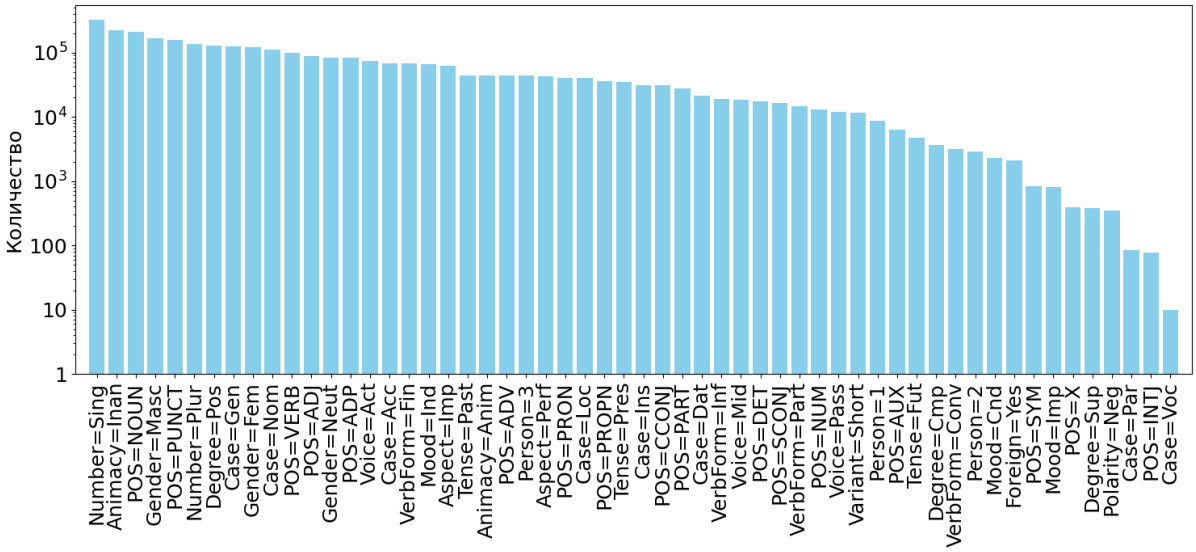

Рис. 7: Распределение граммем, GSD версии 2.1

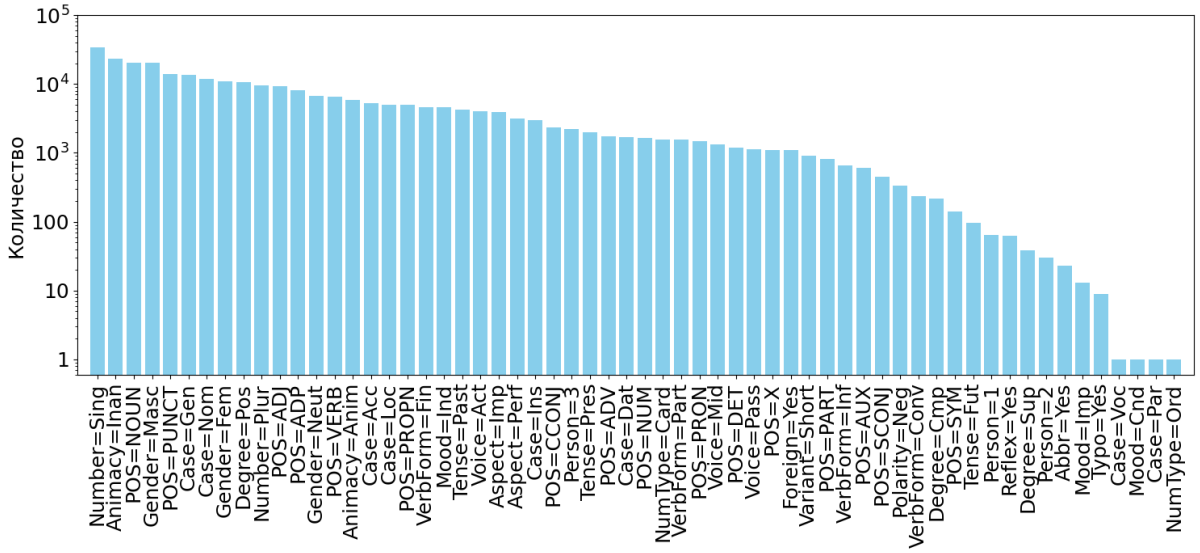

Рис. 8: Распределение граммем, GSD версии 2.15

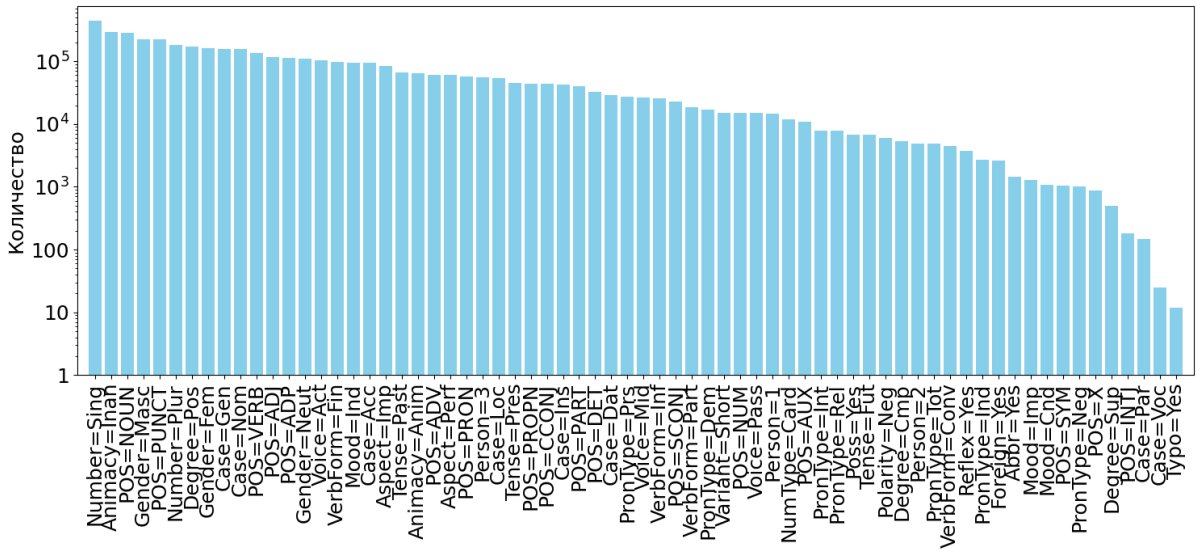

Рис. 9: Распределение граммем, GSD версии 2.15

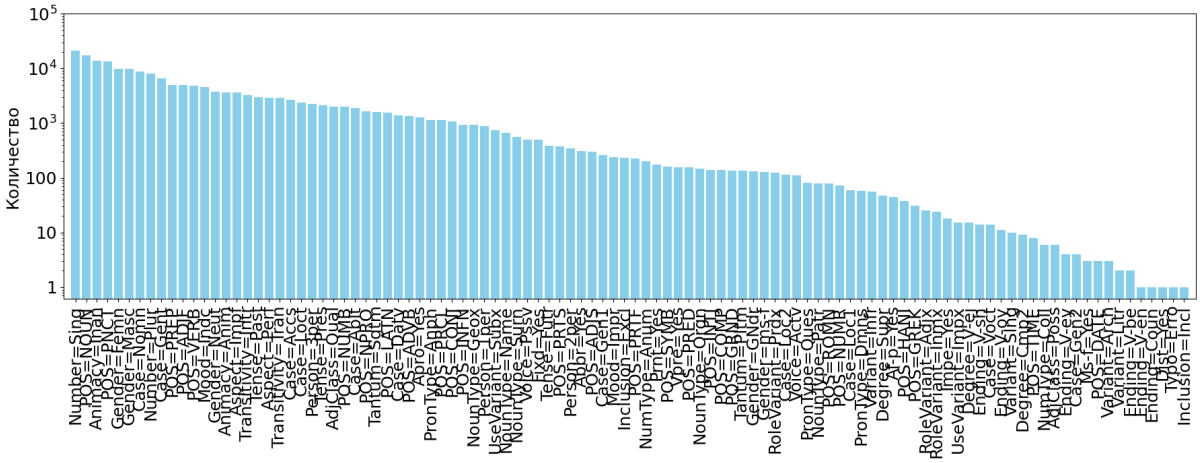

Рис. 10: Распределение граммем, OpenCorpora

Приложение Б

Статистика распределения категорий по корпусам

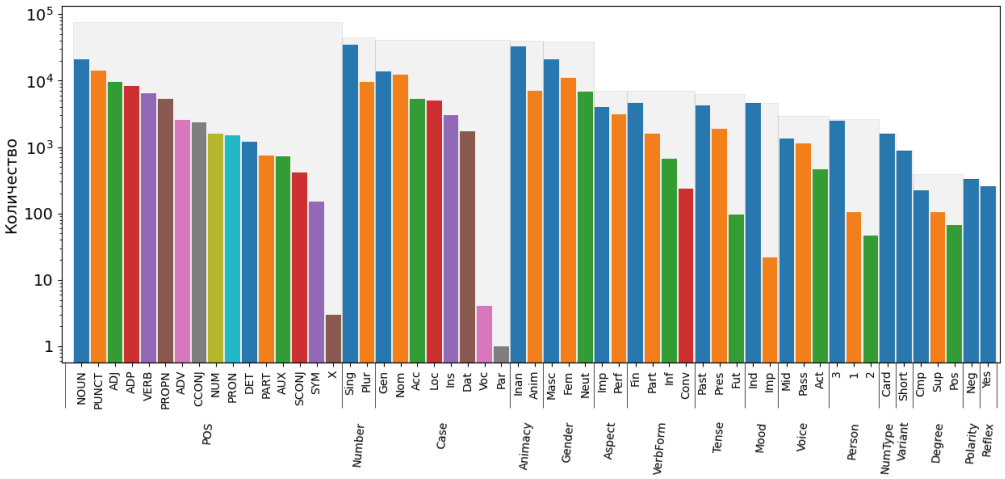

Рис. 11: Распределение категорий, GSD версии 2.1

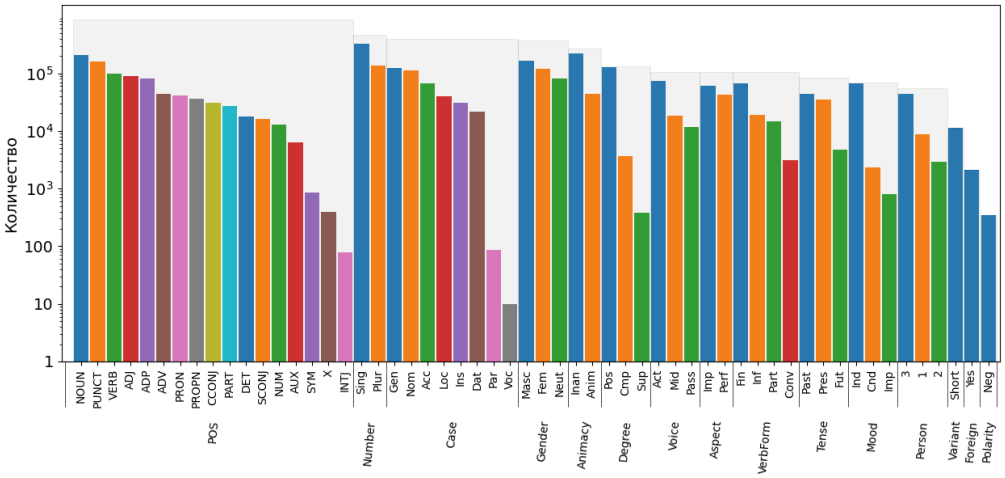

Рис. 12: Распределение категорий, GSD версии 2.1

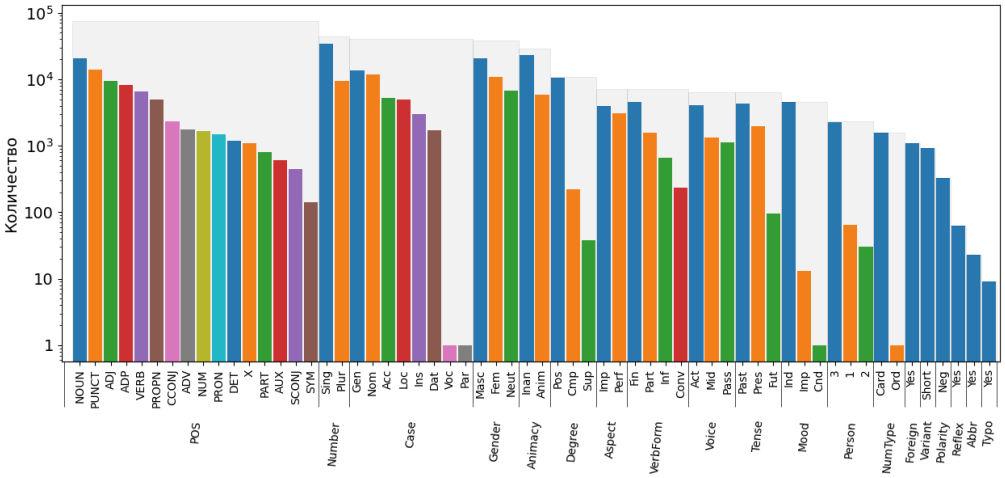

Рис. 13: Распределение категорий, GSD версии 2.15

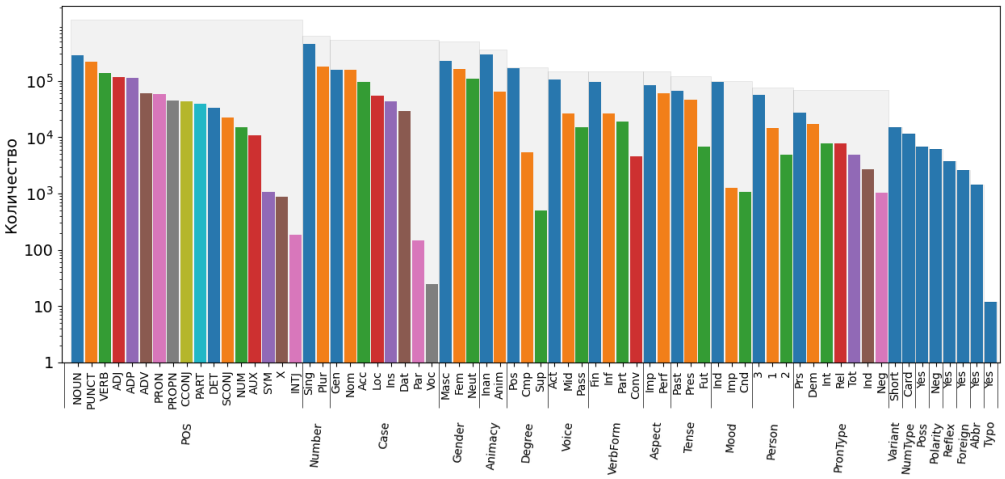

Рис. 14: Распределение категорий, GSD версии 2.15

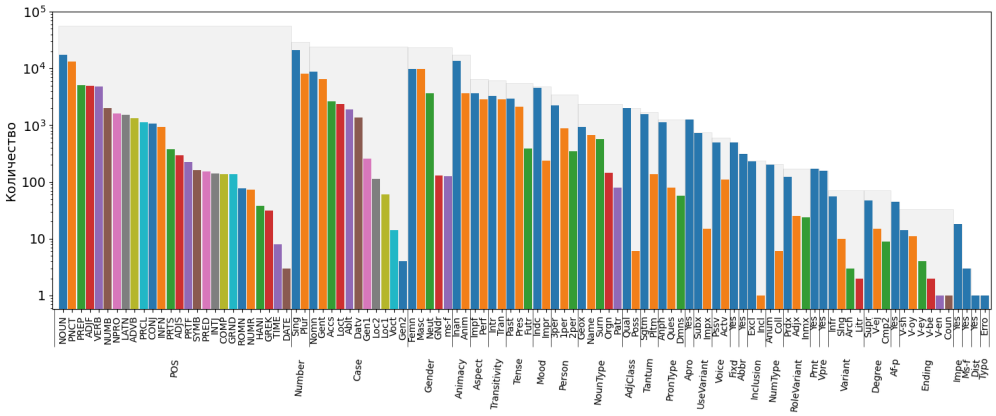

Рис. 15: Распределение категорий, OpenCorpora