Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский физико-технический институт (государственный университет)»

Факультет инноваций и высоких технологий
Кафедра анализа данных

Направление подготовки: 01.03.02 Прикладная математика и информатика

Автоматическая расстановка ударений в словах

Бакалаврская работа

Обучающийся: Захаров Александр Сергеевич

_____

Научный руководитель: Конушин Антон Сергеевич
к.ф.-м.н., доцент

_____

Москва 2018

Аннотация

Ударение является важным элементом устной и письменной речи. В русском языке их расстановка по написанию слова является сложной задачей, а ее решение необходимо для синтеза и распознавания речи. В работе исследуется нейросетевой подход к решению задачи постановки ударений. Было опробовано несколько различных архитектур на основе рекуррентых нейронных сетей. Было исследовано символьное и слоговое представления данных. В работе была исследовано влияние размера обучающей выборки на качество работы модели.

Удалось добиться accuracy score 0.985 путем использования модели с attention со слоговым представлением данных. Был подобран оптимальный размер обучающей выборки.

Оглавление

Введение

Ударение в словах – важнейший элемент устной, письменной и внутренней речи. В русском языке оно играет исключительно важную роль, так как благодаря ему мы можем различать слова. Одной из сложностей русского языка является его свободное ударение, которое не закреплено за каким-либо определенным слогом или морфемой слова. Любой слог может выделяться фонетически. К тому же ударение может меняться с изменением грамматической формы слова. Как отмечает лингвист Н. А. Еськова, «слова с подвижным ударением в русском языке исчисляются сотнями. В процентном отношении это немного, но среди них много чрезвычайно употребительных, поэтому в речи они достаточно заметны» [1]. Например: фла́г — фла́га — фла́ги; но вра́г — врага́ — враги́

Есть языки, где ударение всегда на одном и том же слоге — такое ударение называют фиксированным. Например, во французском ударение всегда на последнем слоге, в польском — на предпоследнем, в чешском — на первом. В русском языке аналогичные правила весьма размыты, поэтому если человек не знает, как правильно ставить ударение в слове, то по одному только его внешнему облику сделать правильный выбор бывает сложно. Нет общих правил ударения и в заимствованных для русского языка словах . Иногда оно меняет свое место по сравнению с ударением в языке-источнике: ноутбу́к, скелето́н, футбо́л, хокке́й. А иногда сохраняет: бульо́н, гардеро́б, жалюзи́. Расстановка ударений, как часть задачи предсказания произношения, - важная составляющая приложений, таких как: автоматическое распознавание речи, синтез речи, транслитерация. Кроме того – это необходимо всем, изучающим русский язык.

Глава 1.
Литературный обзор

Работы по предсказанию постановки ударений в словах велись в двух направлениях. На основе лингвистических правил [2, 3] и на основе анализа данных, где модели строятся напрямую из текстов с обозначенными ударениями.

В русском языке сохраняется множество индо-европейских шаблонов ударений. Чтобы узнать ударение морфологически сложного слова, состоящего из основы и окончания, необходимо узнать является ли основа ударной и на какой слог падает в ней ударение, либо ударным является окончание [4].

1.1. Предсказание ударения в слове на основе ранжирования

Авторы [5] рассмотрели проблему расстановки ударений, как задачу ранжирования. В своем исследовании они опирались на более раннюю статью [6]. Из каждого слова выделяются гласные буквы и они предполагаются, как возможные варианты постановки ударений. Целью модели является отранжировать варианты так, чтобы верная гипотеза имела наименьший ранг.

Для ранжирования гипотез применялось Maximum Entropy ранжирование [7]. Во время обучения модели ей подавался набор правильных гипотез и их признаков. Во время предсказания в модель подавались все гипотезы, и в качестве верной выбиралась гипотеза с максимальным предсказанным результатом. В качестве основы для ранжирования использовалась линейная модель, вместо SVM, так как она более эффективна с вычислительной точки зрения для обучения и применения.

В базовой статье [6] признаками являлись триграммы для гласных букв следующего вида: предыдущая согласная, если она есть, гласная буква, следующая за ней слогласная, если она есть (Dou). На основе лигвистического исследования в данной статье авторы добавили следующие признаки: для каждого слова взяты все начальные и конечные части (уже - у, уж, уже, е, же) (Affix). Также эти признаки добавлены в следующем виде: все буквы заменены на их абстрактные фонетические классы (представлены в табл. 1.1)(Abstr Aff).

Таблица 1.1 — Абстрактные фонетические классы

Класс	Буквы
vowel	а, е, и, о , у, э, ю, я, ы
stop	б, д, г, п, т, к
nasal	м, н
fricative	ф, с, ш, щ, х, з, ж
hard/soft	ъ, ь
yo	ё
semivowel	й, в
liquid	р, л
affricate	ц, ч

В качетсве данных авторы использовали Грамматический словарь русского языка Зализняка[8], разбитый на обучающую и тестовую выборки. Из тестовой выборки также были отдельно выделены те слова, которые не встречались в обучающей выборке, и для них также были получены результаты. Результаты экспериментов представлены в табл. 1.2.

Таблица 1.2 — Результаты ранжирования

Признаки	Accuracy score
Тестовая выборка	
Dou	0.972
Aff	0.987
Aff+Abstr Aff	0.987
Dou et al+Aff	0.987
Dou et al+Aff+Abstr Aff	0.987
Слова не встречавшиеся в обучающей выборке	
Dou	0.806
Aff	0.798
Aff+Abstr Aff	0.810
Dou et al+Aff	0.823
Dou et al+Aff+Abstr Aff	0.89

Таблица показывает влияние взаимодействия признаков на обобщающую способность модели, и лучший результат достигнут при использовании всех признаков.

Недостатками этой работы является неиспользование контекста для определения места

ударения. При подсчете результатов никак не учитывалась частота употребления слов в текстах языка, использовался просто его лексический набор.

1.2. Предсказание ударения в слове при помощи конечного преобразователя

Целью авторов[9] была разработка модели, которая могла быть помочь людям изучать русский язык. Авторы решили, что в некоторых словах ударение может быть пропущено. Считалось, что неправильное ударение может быть хуже, чем его отсутствие для человека, осваивающего новый язык.

Модель состояла из двух частей: конечного преобразователя [10, 11], который из полученного слова генерировал все возможные, корректные по его мнению, позиции ударения. Далее при помощи формальной грамматики [12] удалялись варианты, которые не подходили по контексту. Если после применения этой процедуры оставался один вариант прочтения, то он и выбирался как финальный. Если ни одного – то ударение в слове не проставлялось. Если же вариантов было несколько, – то в зависимости от эксперимента выбиралась дальнейшая стратегия.

Авторами использовался корпус текстов, состоящий из 7689 слов с размеченными ударениями, это были тексты для начинающих изучать русский язык. Также для обучения модели применялся Грамматический словарь русского языка Зализняка [8].

Описание экспериментов:

- **bare:** при нескольких возможных вариантах прочтения слова, ударение в слове не проставлялось.
- **safe:** при нескольких возможных вариантах прочтения, ударение в слове выставлялось, если во всех них ударение падало на один и тот же слог.
- **randReading:** при нескольких возможных прочтениях, случайно выбиралось одно с вероятностью выбора варианта равной частоте встречаемости этого варианта в тексте.
- **freqReading:** при нескольких возможных прочтениях, выбирается вариант с максимальной частотой встречаемости среди всех вариантов в тексте.

Эксперименты были проведены при использовании формальной грамматики с учетом контекста и без него. Результаты представлены в табл. 1.3. Для слов, которые не встретились в словаре, применялось простое правило постановки ударения: ударение падает на последню гласную, после которой идет согласная. Это является наиболее вероятным вариантом ударения в русском языке [13]. В результатх это отображено как guessSyl.

Таблица 1.3 — Результаты применения конечного преобразователя

Эксперимент	Accuracy score	Доля ошибок	Доля пропущенных слов
Без грамматики			
bare	30.43	0.17	69.39
safe	90.07	0.49	9.44
randReading	94.34	3.36	2.30
freqReading	95.53	2.59	1.88
randReading+guessSyll	94.99	4.05	0.96
freqReading+guessSyll	95.83	3.46	0.72
С грамматикой			
bare	45.78	0.44	53.78
safe	93.21	0.74	6.058
randReading	95.50	2.59	1.90
freqReading	95.73	2.40	1.88
randReading+guessSyll	95.92	3.33	0.74
freqReading+guessSyll	96.15	3.14	0.72

При использовании модели без формальной грамматики полнота гипотез составила 97.55%, что является максимумом результата для данной модели. При использовании грамматики полнота составила 97.35%. Эти результаты являются потолком для соответствующих экспериментов. Совмещение всех моделей и предсказываение методом FreqReading позволило получить наибольший процент правильных ударений. Метод расстановки ударений для неизвестных слов в данном случае имеет точность всего 21%. При этом высокая точность была достигнута за счет расстановки ударений почти во всех словах, что соответственно повысило уровень ошибок.

Метод, представленный в этой статье, является попыткой улучшить словарный метод, путем разрешения неоднозначностей в омографах при помощи формальной грамматики. При этом для слов, которые не встретились в словаре, работает очень простой и слабый алгоритм. В этом случае качество получается очень низким. Недостатком является также использование небольшого закрытого корпуса текстов. А так как использовались тексты для начинающих изучать язык, их словарь скорее всего был достаточно мал. Не ясна цель проведения эксперимента RandReading, так как несложно показать строго математически, что метод FreqReading всегда дает большую вероятность правильного ответа.

1.3. Предсказание ударения в слове при помощи символьной нейронной сети

В качестве основы для модели авторами [14] использовалась символьная двусторонняя рекуррентная нейронная сеть на основе LSTM-модулей. На вход подавалась матрица

размера [длина фразы; число возможных символов]. Для кодирования символов было применено one-hot кодирование. Авторами выбрана следующая архитектура: к входной матрице применяется двусторонняя рекуррентная нейронная сеть, сконкатенированные вектора, полученные от рекуррентного слоя подаются в полносвязные слой с softmax активацией. На выходе получается вектор размера равного длине фразы, соответствующий распределению вероятностей постановки ударения в конкретной позиции (рис. 1.1).

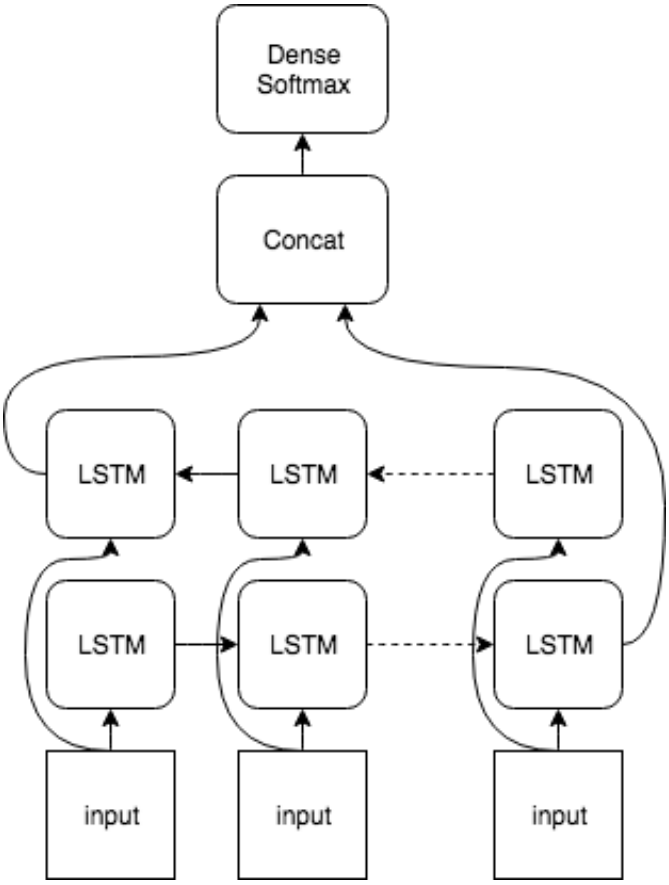

Рисунок 1.1 — Архитектура сети

Для разных экспериментов использовался грамматический словарь русского языка Зализняка [8] и база данных акцентологической разметки в составе национального корпуса русского языка [15].

Авторами были проведены следующие эксперименты.

1) **Обучение и предсказание на основе словаря.** Словарь Зализняка был разделен на обучающую и тестовую выборки в соотношении 2:1.

2) **Обучение и предсказание на основе акцентологического корпуса.** С корпусом было проведено два эксперимента, в первом в качестве фразы использовалось только само слово. Во втором же, к нему были дописаны три последние буквы из слова, которое идет перед ним в предложении, если такое было. Основная разница между моделями с контекстом и без него может быть видна только на омографах. Как видно из

результатов, модель успешно использует контекст для расстовки ударения в омографах во многих случаях.

Результаты этих экспериментов представлены в табл. 1.4.

Таблица 1.4 — Результаты нейросетевой модели

Эксперимент	Модель с контекстом	Модель без контекста
Словарь Зализняка	-	0.751
Текст	0.979	0.977
Омографы в текстах	0.819	0.79

Как видно из представленных данных, нейросетевая модель успешно справляется с использованием контекста для расстановки ударений в омографах. Недостатками же представленной модели является очень простая архитектура и отсутствие работы с текстом. Далее мы будем использовать эту модель как базовую для сравнения результатов.

Глава 2.
Экспериментальная часть

2.1. Используемые данные

Во всех экспериментах в качестве источника данных мы использовали базу данных акцентологической разметки в составе национального корпуса русского языка [15]. С каждым предложением в тексте были произведены следующие преобразования:

1) Все буквы приведены к строчным;
2) Предложение разбито на смысловые подпредложения с использованием в качестве разделителей знаков препинания. После этого все знакия препинания удаляются;
3) Подпредложения, содержавшие иные символы кроме букв кириллицы удалены;
4) По правилам эксперимента из получившихся подпреложений собирались фразы.

Итоговые данные состояли из 3285455 слов. Все данные были разделены на 3 части: обучающую выборку (2299818 слов), валидационную выборку, применяемую для подбора параметров во время обучения модели (49281 слов) и тестовую выборку, на которой измерялся конечный результат (936356 слов).

Омографы среди всех слов в нашей выборке составляют 3.31%. Распределение долей слов по длинам представлено в табл. 2.1. Аналогичное распределение для омографов представлено в табл. 2.2.

Таблица 2.1 — Распределение слов по числу слогов

Число слогов	Доля слов
2	0.474
3	0.308
4	0.144
5	0.053
6	0.015
7	0.003
8	0.001
9	10^{-4}

Таблица 2.2 — Распределение омографов по числу слогов

Число слогов	Доля слов
2	0.736
3	0.212
4	0.051

2.2. Метрики

Основной метрикой, используемой для оценки окончательного качества, является *Accuracy score* (2.1).

Обозначим позицию ударения во фразе как y_i. Позицию ударения, предсказанную моделью, обозначим как y_i^*. Число фраз в выборке обозначим как N

$$ACC = \frac{\sum_{i=1}^{N} I\left\{y_i = y_i^*\right\}}{N} \tag{2.1}$$

2.3. Локальная модель

В качестве самой простой нейросетевой архетектуры нами была выбрана данная архитектура. Ее можно считать упрощением базовой модели [14].

2.3.1. Архитектура модели

К входным данным применяется двустронняя рекуррентная нейронная сеть на основе LSTM-модулей. Далее к каждому промежуточному вектору применяется один и тот же полносвязный слой с softmax активацией с размерностью выхода 2. Первую компоненту этого вектора мы интерпретируем, как вероятность того, что в данной позиции нет ударения, вторую – как то что оно есть (рис. 2.1). Эту архитектуру далее мы будем называть локальной моделью.

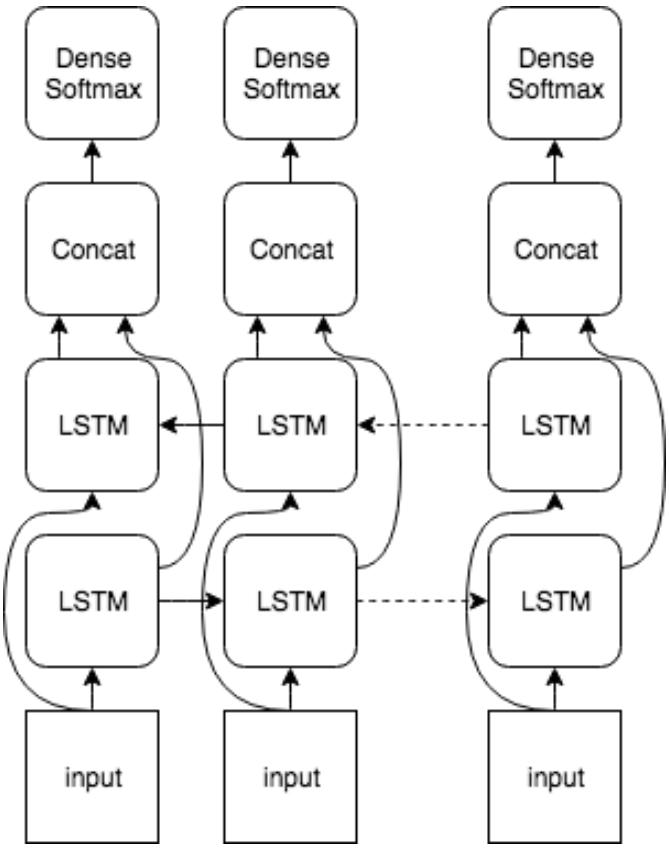

Рисунок 2.1 — Архитектура локальной модели

2.3.2. Посимвольный эксперимент

Входные данные: На вход модели подается слово, в котором мы хотим поставить ударение. Четыре последние буквы предыдущего слова, если оно есть и четыре последние буквы следующего слова, если оно есть. Четыре буквы мы используем, потому что это длина окончания в русском языке. Окончание может нам помочь определить форму слова, что необходимо для удаления неоднозначности при расстановке ударения в большинстве омографов. Пример входных данных и ответа представлен в табл. 2.3

Таблица 2.3 — Пример данных для символьной модели

Фраза	ятой руки было
Матрица ответа	11111111011111 00000000100000

Позиция ударения во фразе выбиралась, как позиция с максимальной вероятностью второго класса.

Результаты этого эксперимента и сравнение с базовой моделью представлены в табл. 2.4

Таблица 2.4 — Результаты локальной и глобальной символьной модели

Число слогов	Локальная модель	Глобальная модель
Все слова		
2	0.961	0.983
3	0.940	0.977
4	0.947	0.976
5	0.960	0.977
6	0.958	0.973
7	0.924	0.955
8	0.866	0.923
9	0.809	0.952
среднее	0.952	0.979
Омографы		
2	0.839	0.810
3	0.774	0.844
4	0.787	0.847
среднее	0.821	0.819

Наше предположение о том, что эта модель более слабая, чем глобальная подтвердилось. При этом, благодаря изменению, заключающемуся в измененииии использования контекста, результаты на омографах удалось улучшить.

2.3.3. Эксперимент с предложением

Для исследования влияния длины контекста на качество расстановки ударений был проведен следующий эксперимент. Контекстом являются не окончания соседних слов, а все подпредложение (часть предложения между знаками препинаниия). Является ли введение в контекст других слов, кроме соседних, значимым, будет ясно по результатам этого эксперимента.

Входные данные: На вход модели подается подпредложение, описание построения которого находится в разд. 2.1. При этом модель должна расставить ударения во всех словах в подпредложении.

Для получения итогового результата из вектора с вероятностями мы в каждом слове выбирали символ с наибольшей вероятностью того, что на него падает ударение. Это, в отличие от отсечения по границе, позволяет добиться того, что в каждом слове находится ровно одно ударение. Пример вхордных данных и матрицы ответа представлен в табл. 2.5

Результаты этого эксперимента и их сравнение с локальной символьной моделью представлены в табл. 2.6.

Таблица 2.5 — Пример данных для модели по предложениям

Фраза	позволяет добиться того
Матрица ответа	11111101111110111111110 00000010000001000000001

Таблица 2.6 — Результаты локальной символьной модели и локальной модели по предложениям

Число слогов	Символьная модель	Модель по предложениям
Все слова		
2	0.961	0.897
3	0.940	0.891
4	0.947	0.902
5	0.960	0.927
6	0.958	0.925
7	0.924	0.898
8	0.866	0.855
9	0.809	0.647
среднее	0.952	0.898
Омографы		
2	0.839	0.831
3	0.774	0.754
4	0.787	0.775
среднее	0.821	0.812

Увеличившийся контекст не привел к повышению результата, при этом из-за усложниившейся задачи качество упало. При этом эта модель требует на порядок больше времени для обучения. А для достижения схожего качества необходимо значительное увеличениие числа LSTM модулей. Поэтому далее мы откажемся от рассмотрения данного подхода к подготовке данных.

2.3.4. Слоговая модель

В русском языке ударение может падать только на гласные буквы. Модели приходилось также учитывать то, что на согласные буквы ударение падать не может. Из-за этого

увеличивалась сложность модели и количество информации, которое она должна хранить. Также из-за этого увеличивалось время обучения.

Одним из вариантов решения этой проблемы является замена символьной модели на слоговую, то есть на вход модели будут подаваться закодированные слоги, а не символы.

Деление на слоги слов в русском языке однозначно установлено [16], поэтому преобразование данных будет детерменированно.

После преобразования получилось 14083 слога.

Входные данные: Формат входных данных аналогичен формату, примененному в локальной символьной модели. Пример входных данных и матрицы ответа представлен в табл. 2.7.

Результаты этого эксперимента, а также их сравнение с результатами локальной и глобальной сисвольной моделей представлены в табл. 2.8.

Таблица 2.7 — Пример данных для слоговой модели

Фраза	ля_ет до_би_ться то_го
Матрица ответа	1 1 1 1 0 11 1 1 0 0 0 0 1 00 0 0

Таблица 2.8 — Результаты локальной и глобальной символьной модлелей с локальной слоговой моделью

Число слогов	Слоговая модель	Локальная модель	Глобальная модель
Все слова			
2	0.985	0.961	0.983
3	0.972	0.940	0.977
4	0.972	0.947	0.976
5	0.976	0.960	0.977
6	0.977	0.958	0.973
7	0.947	0.924	0.955
8	0.899	0.866	0.923
9	0.843	0.809	0.952
среднее	0.978	0.952	0.979
Омографы			
2	0.889	0.839	0.810
3	0.832	0.774	0.844
4	0.843	0.787	0.847
среднее	0.877	0.821	0.819

Применение слогового кодирования позволило без изменения архитектуры модели повысить качество расстановки ударений на 2.6% для всех слов. Это позволило этой модели сравняться по качеству с глобальной символьной моделью. При этом качество расстоновки удаления в омографах у этой модели выше. Из этого всего можно сделать вывод, что обучение модели на слогах вместо символов может помочь повысить результат без изменения архитектуры. Это можно объяснить тем, что модель учит векторные представления для слогов отдельно в embedding слое, а не пытается выучить подобные взаимосвязи в рекуррентном слое. Или в слоге содержится больше информации, чем в одной букве.

2.4. Глобальная модель

2.4.1. Архитектура модели

Архитектура этой модели совпадает с архитектурой базовой модели. Подробно она описана в разд. 1.3. Схема архитектуры представлена на рис. 2.2

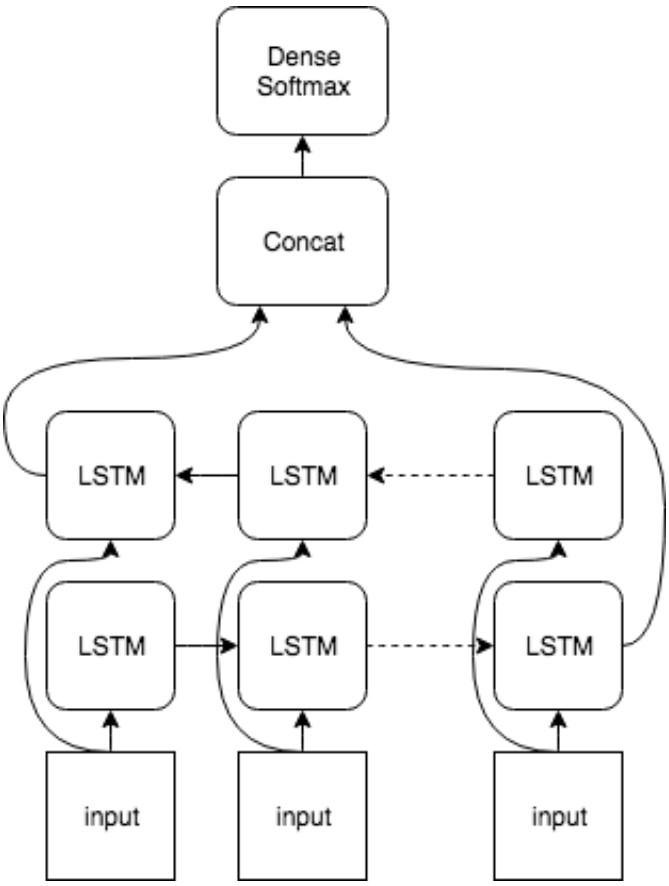

Рисунок 2.2 — Архитектура глобальной модели

2.4.2. Символьная модель

Эта модель является нашей базовой. Опиисание данных и результаты этой модели представлены выше в литературном обзоре в разд. 1.3

2.4.3. Слоговая модель

Использование слогов, как базовой входной единицы модели показало положительный результат при использовании с локальной моделью. Мы рассчитываем также получть положительный результат с глобальной моделью.

Результаты этого эксперимента, и их сравнение с глобальной символьной моделью и локальной слоговой моделью представлены табл. 2.9.

Таблица 2.9 — Результаты локальной и глобальной слоговой модлелей с глобальной символьной моделью

Число слогов	Глобальная модель	Локальная модель	Символьная модель
Все слова			
2	0.985	0.985	0.983
3	0.978	0.972	0.977
4	0.977	0.972	0.976
5	0.977	0.976	0.977
6	0.970	0.977	0.973
7	0.945	0.947	0.955
8	0.895	0.899	0.923
9	0.849	0.843	0.952
среднее	0.981	0.978	0.979
Омографы			
2	0.893	0.889	0.810
3	0.847	0.832	0.844
4	0.852	0.843	0.847
среднее	0.882	0.877	0.819

В глобальнной модели использование слогового кодирования также дало повышение результата по сравнению с символьной моделью. Поэтому далее символьные модели мы больше не будем рассматривать. Также это первая модель, которая показала лучший результат, чем базовая модель.

Глобальная модель по сравнениию с локальной имеет большие проблемы с расстановкой ударения в словах, содержащих более чем из 7 слогов. Это можно объяснить тем, что такие слова очень редко встречаются, и полносвязный слой выучивает, что последние слоги почти всегда безударные. Глобальной модели нужен гораздо более сильный сигнал, для постановки ударения в последний слог по сравнению с локальной. В локальной же модели полносвязный слой применяется в каждой позиции независимо, поэтому зависимость от позиции менее выражена. В символьной модели этот эффект меньше проявлен, так как длина слова в слогах и буквах не связана линейно, поэтому более длинные слова там встречаются чаще.

2.5. Модель с attention слоем

Значительного увеличения качества в машинном переводе удалось добится благодаря замене encoder-decoder архитектуры на основе только рекуррентного слоя добавление attention механизма.

2.5.1. Описание attention механизма

Описание attention мехаизма производится на основе одной из первых статей, где он был представлен [17]. Описанный далее attention механизм также известен как soft attention.

Целью attention-механизма является предсказание вектора текущего сотояния(a_i), который будет проинтерпретирован в соответствие с поставленной задачей. При этом на вход ему подается входной вектор состояния (v_i), вектора промежуточных состояний кодирующего слоя ($h_t, t \in [0; l)$, где l - длина входных данных), чаще всего это слой на основе рекуррентных модулей. Сначала для каждого промежутного вектора вычисляется его значимость по отношение к текущему состоянию (2.2). Далее полученные значимости преобразуются в веса для каждой позиции $\alpha_{i,t}$ при помощи sotmax преобразования (2.3). Взвешенная сумма промежуточных векторов h_t с весами $\alpha_{i,t}$ называется контекстным вектором c_i 2.4. Выходной вектор получается из контекстного линейным преобразованием и затем покоординатным нелинейным преобразованием. В данном случае – это гиперболический тангенс (2.5).

$$score(h_t, v_i) = v_i^T tanh\left(W_1 v_i + W_2 h_t\right) \tag{2.2}$$

$$\alpha_{i,t} = \frac{\exp\left(score\left(h_t, v_i\right)\right)}{\sum\limits_{k=0}^{l-1} \exp\left(score\left(h_k, v_i\right)\right)} \tag{2.3}$$

$$c_i = \sum_{t=0}^{l-1} \alpha_t h_t \tag{2.4}$$

$$a_i = tanh\left(W_3 c_i\right) \tag{2.5}$$

В представленных выше уравнениях W_i - матрицы линейных преобразований.

Благодаря подсчету весов в каждом промежуточном состоянии можно визуализировать то, какие части входа были наиболее важными для получения результата в конкретной позиции. При использовании attention механизма в нейросетевом машинном переводе такая визуализация получается хорошо интерпретируемой: для перевода текущего слова наиболее важными являются оригинальные слово или слова, которые передают его смысл. Пример при переводе с французского языка на английский язык представлен на рис. 2.3

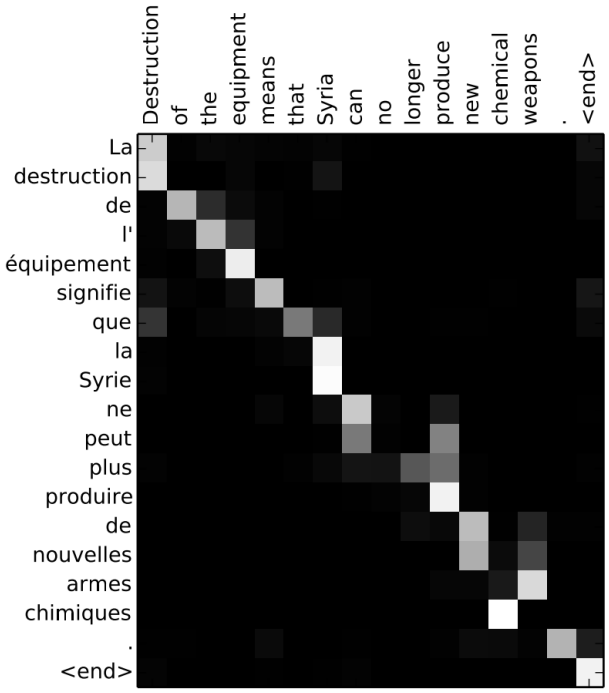

Рисунок 2.3 — Визуализация весов $\alpha_{i,t}$ при переводе

2.5.2. Архитектура модели

Так как нам нужно предсказать только вектор распределения вероятности ударения в конкретной позиции, мы не будем использовать декодирующую сеть, а сразу однократным применением attention слоя получим искомый вектор.

Модель состоит из двустороннего рекуррентного слоя на основе LSTM модулей. К выходному вектору рекуррентного слоя применяется полносвязный слой, полученный вектор будет рассматривать как вектор текущего состояния для attention слоя. Далее следует attention слой и к полученному на его выходе вектору применяется полносвязный слой с softmax активацией.

Схема этой архитектуры представлена на рис. 2.4

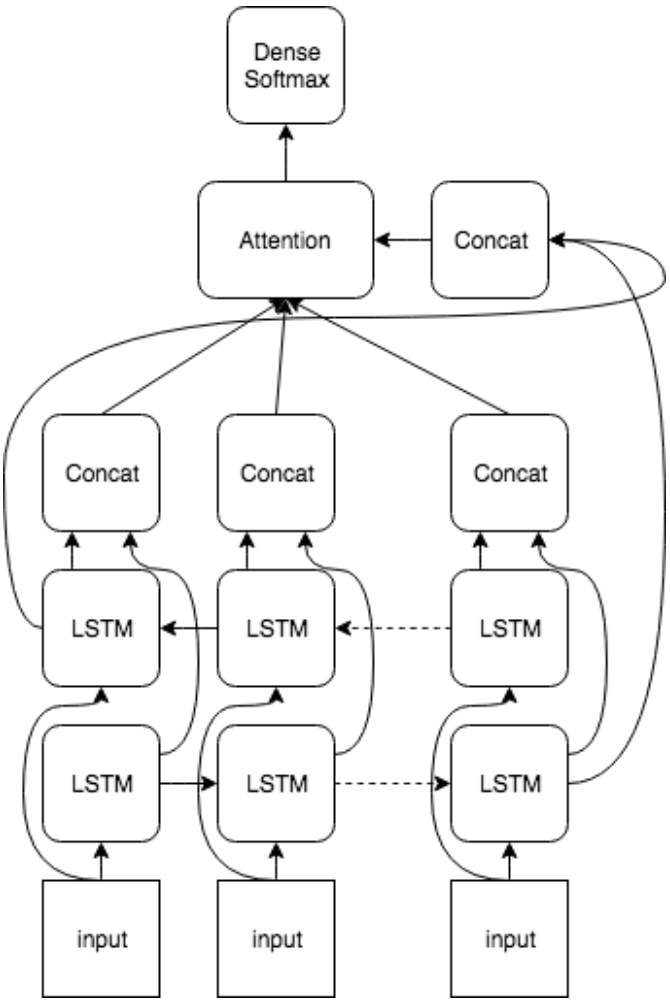

Рисунок 2.4 — Архитектура сети с attention

Для нашей модели примение attention механизма можно считать отчасти объединениием локального и глобального подхода: мы используем все промежуточные состояния рекуррентного слоя, как в локальной модели. При этом предсказываем сразу распределение вероятностей постановки ударения в конкретную позицию, как в глобальной модели.

2.5.3. Слоговая модель

Слоговое кодирование показало себя лучше в других моделях, что было показано выше. Поэтому модель с attention обучалась только в таком режиме. Сравнение результатов с глобальной слоговой моделью представлено в табл. 2.10

Таблица 2.10 — Результаты слоговой глобальной модели и модели с attention

Все слова		
Число слогов	Глобальная модель	Модель с attention
2	0.985	0.989
3	0.978	0.982
4	0.977	0.979
5	0.977	0.980
6	0.970	0.969
7	0.945	0.936
8	0.895	0.867
9	0.849	0.747
среднее	0.981	0.985
Омографы		
2	0.893	0.900
3	0.847	0.869
4	0.852	0.846
среднее	0.882	0.889

Архитектура с attention дает улучшениие по сравнению с глобальной моделью. Однако проблемы глобальной модели со словами из большого числа слогов проявляются еще более сильно. Однако это является лучшим результатом, полученным нами.

2.6. Анализ слогов

Также интересным объектом для исследования являются векторные представления слогов, которые выучивает модель. Далее показана проекция в двумерное пространство, выполненая методом t-SNE, векторных представлений слогов, полученных от модели с attention(рис. 2.5).

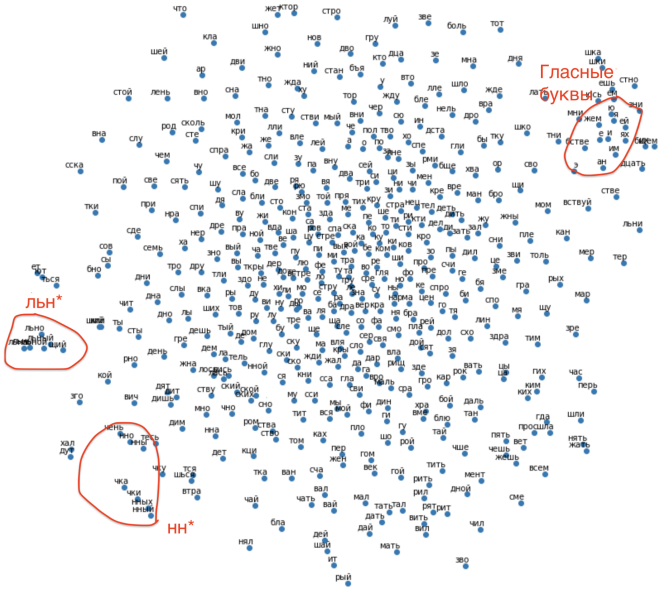

Рисунок 2.5 — Проекция векторных представлений слогов

Кластерной структуры в слогах выделить неудается. Но можно выделить несколько групп по краям. Есть немало слогов, находящихся близко, имеющих большие общие подстроки. Например: вна, вну вны, вно. Из этого можно сделать вывод, что число слогов избыточно и уменьшив размер алфавита нам удасться выделить более мелкие осмысленные элемнты. Как это сделать будет рассмотрено ниже.

2.7. Эксперименты с BPE токенизацией

Одной из идей получения подслов, используемой в машинном переводе, является BPE токенизация.

2.7.1. Описание BPE токенизации

С текстовым корпусом с изначальным алфавитом из букв проводятся следуюющие преобразовния [18]:

1) К концам всех слов дописывается специальный символ <we>;
2) Для всех пар соседниих символов подсчитывается число появлений в корпусе;
3) Самая часто встречаемая пара добавляется в алфавит как новый символ и все ее появления в корпусе заменяется на него;
4) Если размер алфавита меньше желаемого – вернуться к пункту 2.

2.7.2. Применение BPE токенизации к слогам

В нашей задаче в качестве отдельных слов мы будем рассматривать слоги. Размер алфавита был выбран 1000, так как при таком размере многие части слогов еще представляют собою отдельные символы, а не слоги. Объединение в слоги произошло бы при слишком большом размере алфавита, приближающемся к числу слогов.

На таких данных была обучена модель с attention. К сожалению, не удалось добиться повышения результатов по сравнению с моделью со слогами. Получили ухудшение результата. Это представлено в табл. 2.11. Поэтому от этой идеи пришлось отказаться.

Таблица 2.11 — Результаты модели с attention при работе со слогами и BPE токенами

Тип слов	Слоговая модель	Модель с BPE
Все слова	0.985	0.981
Омографы	0.900	0.853

2.8. Анализ ошибок

Анализ ошибок будет проводится на основе нашей лучшей модели: слоговая модель с attention.

2.8.1. Анализ влияния контекста

Контекст является необходимым для определение ударения в омографах.

Для анализа его влияния возьмем фразы из тестовой выборки, которые имеют и левый, и правый контексты. Попытаемся проставить в них ударение в следующих случаях: с двумя контекстами, только с левым, только с правым и без контекста. Результаты этого эксперимента преставлены в табл. 2.12.

Таблица 2.12 — Результаты расстановки ударения с разными типами контекста

Тип контекста	Accuracy score
Левый и правый	0.986
Левый	0.984
Правый	0.977
Без контекста	0.976

Наличие левого контекста очень сильно влияет на результат предсказания. При этом правый контекст также повышает качество, но гораздо слабее.

Как говорилось выше, для модели с attention можно построить распределение весов для каждого элемента входных данных. Входную фразу можно разделить на следующие части: левый контекст, разделитель, слово, разделитель, правый контекст. Эти веса можно интерпретировать как важность этой части фразы для получения итогового результата. Такое распределение представлено для фраз, содержащих оба контекста в табл. 2.13

Таблица 2.13 — Распределение весов attention слоя по частям фразы

Тип слов	Левый контекст	Левый пробел	Слово	Правый пробел	Правый контекст
Все слова	0.008	0.294	0.551	0.140	0.005
Омографы	0.014	0.455	0.391	0.130	0.007

Видно, что вес разделителей очень высок, так как они являются ограничением на область выставления ударения, а также они разделяют семантически разные части фразы. В омографах вес левого контекста гораздо больше, чем в среднем по выборке. Про правый контекст такого вывода сделать нельзя, так как различия в значениях очень малы. Также вес самого слова в омографах ниже.

Более высокий вес слова в обычных словах, по сравнению с омографами, можно объяснить тем, что в обычном слове его самого достаточно для простановки ударения, а в омографах контекст необходим для однозначной постаноки ударения.

Из этого мы можем сделать вывод о том, что левый контекст наиболее важен при растановке ударенииий, роль правого контекста гораздо ниже.

2.8.2. Работа модели с омографами

Омографы являются наиболее сложными словами для расстановки ударений, так как ударение в них зависит не только от их написания, но и от контекста. Для нашей модели с учетом того, что таких слов всего 5% в текстах на них совершается 15% ошибок. У всех омографов всего 2 варианта ударения. При этом для слов, у которых доля одного выриантa выше 75-80%-ов, всегда предсказывалось именно оно, и модели не удавалось определить омографную природу этого слова.

В русском языке омографы можно раделить на несколько групп: слова разных частей речи (уже́ - у́же), словоформы одного слова (руки́ - ру́ки), и слова одной и той же части речи (за́мок - замо́к). Наиболее хороший результат наша модель показала при работе с омографами из разных частей речи (94.3%). Для словоформ результат хуже и составил (84.7%). Третья же группа омографов не может быть корректно обработана нашей моделью, так как левые и правые контексты таких слов совпадают.

2.8.3. Работа модели со словами, которых не было в обучающей выборке

Уникальных слов, которых не было в обучающей выборке встретилось 9805. Accuracy score: 0.838. Ошибки можно разделить на следующие группы:

1) заимствованнные слова, в которых правила ударения отличатся от правил русского языка;

2) русские фамилии, в которых ударение совпадает с формой родительного падежа, но при этом они сами стоят в именительном;

3) многоосновные слова, где модель не может выбрать, на какой корень должно ставиться ударение.

2.9. Active learning

Отдельной темой для исследования является то, какого числа данных достаточно для достижения сходного качества. Техникой отбора данных, которая работает даже для неразмеченных данных, является Active learning [19].

Идея заключается в том, что модели подается изначальный набор обучающих данных. На нем модель обучается. Из оставшихся данных выбираются примеры, в которых модель наиболее неуверена 2.6. Эти данные добавляются к обучающей выборке. И так продолжается до тех пор, пока качество не перестанет меняться от добавления новых данных. Данные выбираются по схеме выбора без возвращения.

$$P = \prod_{i=1}^{l} \max(p_i, 1 - p_i) \tag{2.6}$$

В нашем случае отдельно имеются омографы, на которых модели сложнее работать. Поэтому мы провели несколько разных экспериментов с active learning: в стартовых данных 100% омографов, 50%, 5% и вообще без омографов. Для повышения скорости обучения эксперименты проводились с глобальной слоговой моделью.

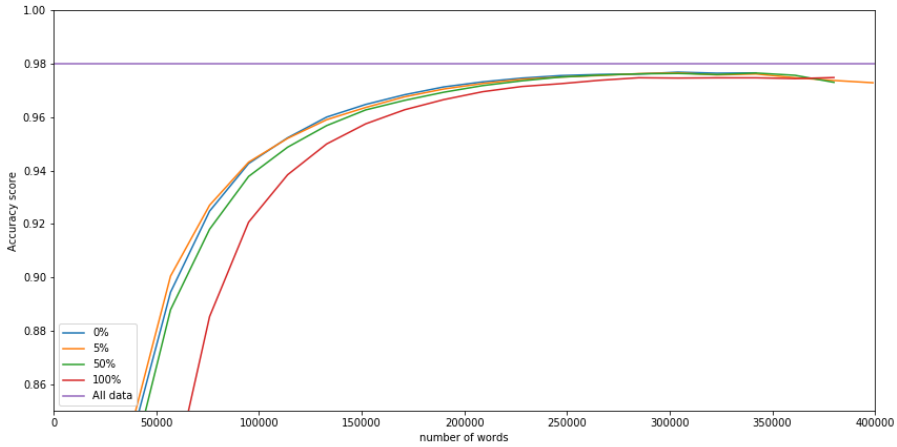

Рисунок 2.6 — Результаты Active learning на всей тестовой выборке

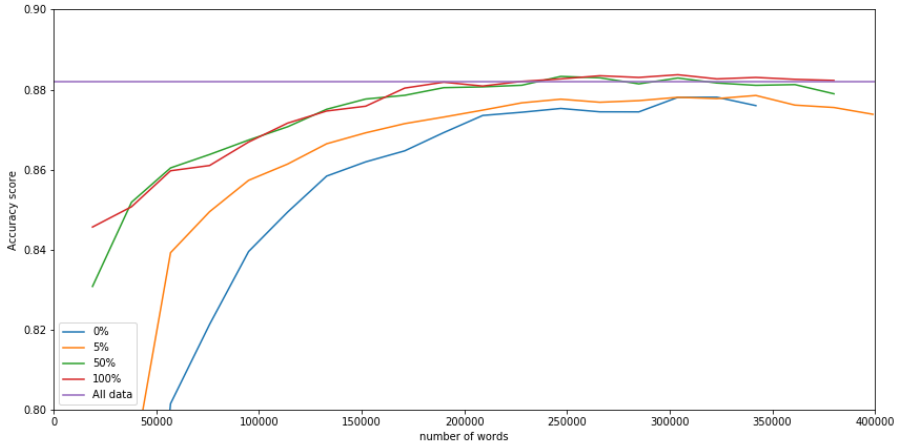

Рисунок 2.7 — Результаты Active learning на омографах

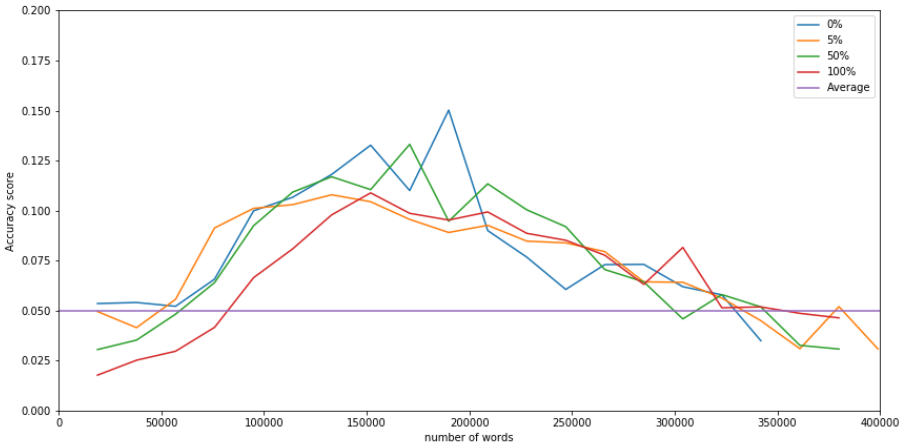

Рисунок 2.8 — Доля взятых омографов на каждом шаге

Как видно на рис. 2.6 во всех случаях удалось добиться схожего качества с моделью, обученной на всех данных, кроме экстремального эксперимента со стартовыми данными, состоящимии только из омографов. При этом: чем больше процент омографов в стартовой выборке, тем выше получается качество расстановки ударений в них (рис. 2.7). В середине обучения во всех экспериментах модели начинали собирать больше омографов в обучающую выборку, чем их среднее число в тексках. Это можно объяснить тем, что модель выучивает основные правила и начинает сомневаться в омографах (рис. 2.8]).

Такого качества удалось добиться на 20% исходных данных. Важным вопросом при этом является сравнение active learning со случайный выбором примеров. Такое сравнение представлено в табл. 2.14

Таблица 2.14 — Сранение Active learning со случайным выбором примеров

Тип слов	Active learning	Случайный выбор
Все слова	0.976	0.960
Омографы	0.883	0.863

Качество при использованиии Active leaning получилось выше, чем если обучать модель на случайных 20% обучающих данных. Благодаря особому методу отбора примеров, нам удалось добиться более высокого качества на меньшем объеме данных.

Заключение

Итоги работы

В данной работе мы подробно исследовали примененииe нейросетевого подхода для задачи расстановки ударений в русском языке. Основными итогами нашей работы являются:

1) Проведено сравнение несколько нейросетевых архитектур моделей для расстановки ударений в русском языке. Лучшей архитектурой получилась модель с attention.

2) Улучшено качество расстановки ударений с 0.979 до 0.985.

3) Исследовано влияние представлений входных данных накачество работы модели.

4) Исследовано влияние контекста на качество расстановки ударений.

5) Определен эффективный размер обучающей выборке при помощии Active learning и он составил 20% от всех обучающей выборки.

Дальнейшие исследования

Идеи для дальнейших исследований можно разделить на несколько групп:

1) **Архитектура модели.** Использованная нами финальная модель с attention является попыткой адаптировать универсальную модель. Использование более продвинутой универсальной модели или построение специализированной может обеспечить повышение качества. Также можно добавить модели на вход дополнительную информацию, например, морфологические признаки слов.

2) **Работа с омографами.** Текущая модель хорошо работает с омографами, которые являются разными частями речи или разными формами одного слова. Хотелось бы добавить обработку омографов, которые являются одной и той же частью речи и при этом имеют одну и ту же форму.

3) **Работа с именованными сущностями.** Наша модель плохо справляется с расстановкой ударений в именованных сущностях, из которых можно отдельно выделить группу русских фамилий. Для этих слов, например, можно построить отдельную модель.

Список использованных источников

[1] Еськова Н.А. Словарь трудностей русского языка. Ударение. Грамматические формы. *М.: Языки славянской культуры*, 2014.

[2] Kenneth Church. Stress assignment in letter to sound rules for speech synthesis. *Association for Computational Linguistics, 246–253*, 1985.

[3] Briony Williams. Word stress assignment in a text-to-speech synthesis system for british english. *Computer Speech and Language, 2:235–272*, 1987.

[4] Morris Halle. On stress and accent in indoeuropean. *Language*, 1997.

[5] Keith Hall and Richard Sproat. Russian stress prediction using maximum entropy ranking. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 879–883, Seattle, Washington, USA*, 2013.

[6] Qing Dou, Shane Bergsma, Sittichai Jiampojamarn, and Grzegorz Kondrak. A ranking approach to stress prediction for letter-to-phoneme conversion. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 118–126, Suntec, Singapore*, 2009.

[7] Michael Collins and Terry Koo. Discriminative reranking for natural language parsing. *Computational Linguistics, 31:25–69*, 2005.

[8] Зализняк А. А. Грамматический словарь русского языка. *М.: Русский язык*, 1977.

[9] Robert Reynolds and Francis Tyers. Automatic word stress annotation of russian unrestricted text. *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015). Linkoping University Electronic Press, Sweden, Vilnius, Lithuania*, 2015.

[10] Kimmo Koskenniemi. Two-level morphology: A general computational model for word-form recognition and production. *Technical report, University of Helsinki, Department of General Linguistics*, 1983.

[11] Kenneth R. Beesley and Lauri Karttunen. Finite state morphology: Xerox tools and techniques. *CSLI Publications, Stanford*, 2003.

[12] Fred Karlsson. Constraint grammar as a framework for parsing running text. *Proceedings of the 13th Conference on Computational Linguistics (COLING), 3:168–173, Helsinki, Finland*, 1983.

[13] Yulia Lavitskaya and Baris Kabak. Phonological default in the lexical stress system of russian: Evidence from noun declension. *Lingua*, 2014.

[14] Maria Ponomareva, Kirill Milintsevich, Ekaterina Chernyak, and Anatoly Starostin. Automated word stress detection in russian. *Proceedings of the First Workshop on Subword and Character Level Models in NLP, 31–35, Copenhagen, Denmark*, 2017.

[15] Гришина Е. А. Корпус «История русского ударения» // Национальный корпус русского языка: 2006—2008. Новые результаты и перспективы. *СПб.: Нестор-История*, 2009.

[16] Литневская Е. И. Русский язык: краткий теоретический курс для школьников. *М.:МГУ*, 2006.

[17] Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *Conference paper at ICLR 2015*, 2014.

[18] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 1715–1725, Berlin, Germany*, 2016.

[19] Yanyao Shen, Hyokun Yun, Zachary C. Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep active learning for named entity recognition. *Conference paper at ICLR 2018*, 2017.